

**UNIVERSITATEA PEDAGOGICĂ DE STAT
„ION CREANGĂ” DIN CHIȘINĂU**

Cu titlu de manuscris
C.Z.U.: 373.091.3:001.2:004(043.3)

CORLAT SERGIU

**ABORDĂRI INTERDISCIPLINARE PENTRU INSTRUIREA DE
PERFORMANȚĂ LA INFORMATICĂ**

**532.02 – DIDACTICĂ ȘCOLARĂ
(PE TREPTE ȘI DISCIPLINE DE ÎNVĂȚĂMÂNT)**

Teză de doctor în științe ale educației

Conducător științific:



Ilie LUPU, doctor habilitat în pedagogie
profesor universitar

Autor:



Sergiu CORLAT

CHIȘINĂU, 2025

© Corlat Sergiu, 2025

CUPRINS

ADNOTARE	4
АННОТАЦИЯ	5
ANNOTATION.....	6
LISTA TABELELOR	7
LISTA FIGURILOR.....	8
LISTA ABREVIERILOR.....	10
INTRODUCERE	11
1. INSTRUIREA DE PERFORMANȚĂ ÎN INFORMATICĂ: CADRUL TEORETIC	24
1.1 Informatica - știință exactă și disciplină școlară.....	24
1.2 Informatica - domeniu competițional de performanță.....	30
1.3 Abordarea interdisciplinară în studiul Informaticii	38
1.4 Concluzii Capitolul 1.....	44
2. MODELUL CRYPTEX PENTRU INSTRUIREA DE PERFORMANȚĂ LA	
INFORMATICĂ: CADRUL CONCEPTUAL ȘI METODOLOGIC.....	46
2.1 Conceptul de instruirea de performanță a elevilor la Informatică.....	46
2.2 Modelul interdisciplinar pentru instruirea de performanță la Informatică - CRYPTEX.....	52
2.3 Specificul aplicării modelului CRYPTEX pentru instruirea de performanță	
la Informatică.....	97
2.4 Profilul elevului performer la Informatică.....	101
2.5 Concluzii Capitolul 2.....	108
3. VALIDAREA MODELULUI INTERDISCIPLINAR DE INSTRUIRE	
PERFORMANȚĂ A ELEVILOR ÎN INFORMATICA COMPETIȚIONALĂ – STUDIU	
DE CAZ	110
3.1 Specificul spațiului și surselor de date validate.....	110
3.2 Analiza comparativă a rezultatelor implementării modelului CRYPTEX de instruire	
versus metode tradiționale	114
3.3 Analiza post – instrucțională	119
3.4 Concluzii Capitolul 3.....	122
CONCLUZII GENERALE ȘI RECOMANDĂRI	124
BIBLIOGRAFIE	128
ANEXE	140
DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII	145
CURRICULUM VITAE A CANDIDATULUI.....	146

ADNOTARE

Corlat Sergiu. Abordări interdisciplinare pentru instruirea de performanță la Informatică Teză de doctor în științe ale educației, Chișinău, 2025

Structura tezei: introducere, trei capitole, concluzii generale și recomandări, bibliografie din 202 de titluri, 2 anexe, 121 pagini de text de bază, 29 de figuri și 9 tabele. Rezultatele obținute sunt publicate în 40 lucrări științifice.

Cuvinte-cheie: disciplina Informatica, curriculum național, olimpiade școlare, performanță informatică, instruire de performanță, model de instruire.

Domeniul de studii: Științe ale educației. Didactica școlară (pe trepte și discipline de învățământ).

Scopul lucrării: fundamentarea teoretică și metodologică a instruirii de performanță a elevilor la disciplina Informatica prin abordări interdisciplinare.

Obiectivele cercetării: 1) Stabilirea valențelor formative a informaticii ca știință și disciplină școlară din perspectiva instruirii de performanță informatică a elevilor; 2) Analiza experienței internaționale în domeniu: cercetarea modelelor de instruire în țările cu tradiții informatice din regiune și a liderilor internaționali; 3) Fundamentarea unui concept de instruire de performanță a elevilor la Informatică din perspectiva competițională; 4) Elaborarea modelului interdisciplinar privind instruirea de performanță a elevilor la disciplina Informatica; 5) Modelarea profilului elevului performer la Informatică; 6) Validarea modelului interdisciplinar privind instruirea de performanță a elevilor la disciplina Informatica.

Noutatea și originalitatea științifică este obiectivată de: Valorificarea valențelor formative a Informaticii ca știință și ca disciplină școlară din perspectiva instruirii de performanță a elevilor; Abordarea interdisciplinară a instruirii de performanță a elevilor la Informatică axată pe principii de integrabilitate a diferitor discipline școlare, în cazul dat: matematica, fizica, chimia, biologia, științele umanistice;

Dezvoltarea competenței elevilor în Informatica competitivă prin complementarea cu noi concepte și operații (unități de competențe); Modelul de instruire performantă a elevilor la Informatică din perspectiva interdisciplinarității, cadrului formativ al disciplinei Informatica, profilului elevului performer la Informatică; Profilul elevului performer la Informatică din perspectiva cognitiv, motivațională, afectivă, comportamentală și socială.

Rezultatele obținute în cercetare care au contribuit la soluționarea problemei științifice rezidă în fundamentarea teoretică și metodologică a instruirii de performanță a elevilor la Informatică din perspectiva competițională la nivel național și internațional.

Semnificația teoretică a cercetării constă în: Dezvoltarea teoriei și metodologiei de predare-învățare a Informaticii prin conceptualizarea instruirii de performanță a elevilor din perspectiva competițională; Descoperirea de noi valențe și oportunități ale Informaticii ca știință și disciplină școlară în dezvoltarea teoriei și practicii de instruire în genere și a instruirii de performanță în speță; Argumentarea valorificării principiului de integrare în structura modelului interdisciplinar de instruire pentru performanță a elevilor la Informatică; Conceptualizarea instruirii de performanță a elevilor la Informatică ca reper pentru elaborarea modelului interdisciplinar de instruire a elevilor din perspectiva competițională; Construcția profilului elevului performer la Informatică ca țintă de atins în cadrul instruirii de performanță la disciplina Informatica.

Valoarea aplicativă a lucrării: modelul scalabil propus pentru instruirea elevilor capabili de performanțe înalte, poate fi aplicat atât în cadrul instruirii tradiționale, cât și în cadrul pregătirii de performanță. Creșterea eficienței învățării se atestă atât în cazul implementării infuzionale, cât și a celei modulare, impactul fiind mai pronunțat în condițiile racordării modelului propus la necesitățile formabililor.

Implementarea rezultatelor științifice s-a realizat în centrul instituțional de instruire pentru performanță în Informatică a rețelei de licee Orizont și în cadrul olimpiadelor de nivel național și internațional.

АННОТАЦИЯ

Corlat Sergiu. Междисциплинарный подход в подготовке учащихся к международным конкурсам по Информатике.

Диссертация доктора педагогических наук, Chişinău, 2025

Структура диссертации: введение, три главы, общие выводы и рекомендации, библиографии из 202 наименований, 2 приложения, 121 страницы основного текста, 29 рисунков и 9 таблиц. Полученные результаты опубликованы в 40 научных работах.

Ключевые слова: дисциплина Информатика, национальная учебная программа, школьные олимпиады, эффективное обучение Информатике, олимпиадный успех, учебная модель.

Область исследования: Педагогические науки. Дидактика школы (по уровням и учебным предметам).

Задачи исследования: теоретическое обоснование и разработка методологии углублённой подготовки учащихся по информатике на основе междисциплинарного подхода.

Цель исследования: 1) Определение формирующих ценностей информатики как науки и учебной дисциплины с точки зрения углублённой подготовки учащихся; 2) Анализ международного опыта в данной области: исследование моделей обучения в странах региона с традициями в конкурсной информатике включая и международных лидеров; 3) Обоснование концепции углублённой подготовки по информатике с точки зрения конкурсной информатики;

4) Разработка междисциплинарной модели углублённой подготовки по дисциплине «Информатика»;

5) Моделирование профиля ученика - олимпийца по информатике; 6) Тестирование междисциплинарной модели эффективного обучения студентов по дисциплине «Информатика».

Научная новизна и оригинальность исследования: Использование образовательных ценностей информатики как науки и как школьной дисциплины с точки зрения углублённой подготовки учащихся; Междисциплинарный подход к углублённой подготовке учащихся по информатике, ориентированный на принципы интегрируемости различных школьных предметов, в данном случае: математики, физики, химии, биологии, гуманитарных наук; Развитие компетентности учащихся в области конкурсной информатики путем дополнения ее новыми понятиями и операциями (единицами компетентности); Модель углублённой подготовки учащихся по Информатике с точки зрения междисциплинарности, образовательная рамка дисциплины «Информатика», Профиль ученика - олимпийца по информатике; Профиль ученика - олимпийца по информатике с когнитивной, мотивационной, аффективной, поведенческой и социальной точек зрения.

Результаты, полученные в ходе исследования, способствовали решению научной проблемы, заключаются в теоретико-методологическом обосновании эффективности подготовки студентов по информатике с точки зрения конкурентоспособности на национальном и международном уровне.

Теоретическая значимость исследования состоит в: Разработке теории и методологии преподавания и изучения информатики путем концептуализации обучения успеваемости учащихся с точки зрения конкуренции; Раскрытии новых значений и возможностей информатики как науки и учебной дисциплины в развитии теории и практики обучения в целом и педагогической подготовки в частности; 3. Аргументации использования принципа интеграции в структуре междисциплинарной модели углублённой подготовки учащихся по информатике; 4. Концептуализации углублённой подготовки учащихся по информатике как ориентира для разработки междисциплинарной модели обучения студентов с точки зрения конкурсной информатики; 5. Создании профиля ученика - олимпийца по информатике как цели, которую необходимо достичь в ходе углублённого обучения по дисциплине «Информатика».

Практическая значимость исследования: предлагаемая масштабируемая модель обучения способных учащихся может применяться как в контексте формального обучения, так и в контексте обучения по результатам в неформальной образовательной среде. Повышение эффективности обучения подтверждается как в случае инфузионных реализаций, так и модульных, причем эффект более выражен в условиях адаптации предлагаемой модели к требованиям обучаемых.

Внедрение результатов исследования осуществлялось в институциональном центре конкурсной информатики в сети лицеев «Оризонт» и в рамках национальных и международных олимпиад.

ANNOTATION

Corlat Sergiu. Interdisciplinary Approaches to Performance Training in Informatics

Doctor thesis in education sciences, Chisinau, 2025

Thesis structure: introduction, three chapters, general conclusions and recommendations, bibliographies of 202 titles, 2 appendices, 121 pages of basic text, 29 figures and 9 tables. The results obtained are published in 40 scientific papers.

Keywords: Computer science, national curriculum, school Olympiads, computer sciences performance, performance instruction, instructional model.

Field of study: Sciences of Education. School didactics (by levels and educational subjects).

Aim of research: theoretical and methodological substantiation of student performance training in the Computer Science discipline through interdisciplinary approaches.

Research objectives: 1) Establishing the formative values of computer science as a science and school discipline from the perspective of computer performance training for students; 2) Analysis of international experience in the field: research of training models in countries with Computer Science traditions in the region and international leaders; 3) Substantiation of a concept of performance training of students in Informatics from a competitive perspective; 4) Development of an interdisciplinary model regarding performance training of students in the discipline of Informatics; 5) Modeling the profile of the performing student in Informatics; 6) Validation of the interdisciplinary model regarding performance training of students in the discipline of Informatics.

Scientific novelty and originality of the research: Capitalizing on the formative values of Informatics as a science and as a school discipline from the perspective of student performance training; Interdisciplinary approach to student performance training in Informatics focused on principles of integrability of different school disciplines, in this case: mathematics, physics, chemistry, biology, humanities; Developing students' competence in competitive Informatics by complementing it with new concepts and operations (units of competence); The model of high-performance training of students in Informatics from the perspective of interdisciplinarity, the formative framework of the Informatics discipline, the profile of the performing student in Informatics; The profile of the performing student in Informatics from the cognitive, motivational, affective, behavioral and social perspectives.

The results obtained in the research which contributed to solving the scientific problem lies in the theoretical and methodological foundation of the performance training of students in Computer Science from a competitive perspective at the national and international level.

Theoretical significance consists of: 1. Developing the theory and methodology of teaching-learning Informatics by conceptualizing the performance training of students from a competitive perspective; 2. Discovering new valences and opportunities of Informatics as a science and school discipline in the development of the theory and practice of training in general and of performance training in particular; 3. Arguing for the valorization of the principle of integration in the structure of the interdisciplinary model of training for the performance of students in Informatics; 4. Conceptualizing the performance training of students in Informatics as a benchmark for the development of the interdisciplinary model of training of students from a competitive perspective; 5. Building the profile of the performing student in Informatics as a target to be achieved within the performance training in the discipline of Informatics.

of the work: the proposed scalable model for training students capable of high performance can be applied both in traditional training and in performance training. The increase in learning **The applicative value** efficiency is attested both in the case of infusional and modular implementation, the impact being more pronounced when the proposed model is aligned with the needs of the trainees.

The implementation of the scientific results was carried out in the institutional training center for performance in Informatics of the Orizont high school network and in the framework of national and international Olympiads.

LISTA TABELELOR

Tabelul	Denumirea	Pagina
Tabelul 1.1	Repartizarea orelor de Informatică și Tehnologii Informaționale (săptămânal) în liceele teoretice și liceele cu profil matematic – informatic în România, Bulgaria și Republica Moldova	Pag. 30
Tabelul 2.1	Syllabus de performanță, completare pentru curriculumul școlar, actualizat 2025	Pag. 48
Tabelul 2.2	Demonstrația creșterii timpului de execuție în cazul algoritmilor exponențiali	Pag. 81
Tabelul 2.3	Dependența timpului necesar pentru determinarea unei parole prin forță brută	Pag. 86
Tabelul 2.4	Timpul necesar pentru generarea secvențelor binare de lungime N	Pag. 86
Tabelul 3.1	Repartiția geografică a centrelor de Performanță Informatică	Pag. 114
Tabelul 3.2	Repartiția instituțională a centrelor de Performanță Informatică, Chișinău	Pag. 115
Tabelul 3.3	Total distincții 2017 - 2025	Pag. 115
Tabelul 3.4	Implicarea absolvenților în activități creative și de cercetare asociate concursurilor de programare	Pag. 121

LISTA FIGURILOR

Figura	Denumirea	Pagina
Figura 1.1	Etapele de desfășurare a olimpiadei naționale la Informatică și pregătire a lotului olimpic pentru concursurile internaționale	Pag. 32
Figura 1.2	Numărul total de participanți la Olimpiada Națională la Informatică. Micșorarea numărului de participanți pentru anul 2022 este motivată de lipsa de la Olimpiadă a premianților anului precedent, acesta fiind considerat 2019	Pag. 34
Figura 1.3	Tipologia problemelor propuse pentru rezolvare participanților la Olimpiada Națională, clasele de gimnaziu, în perioada 2019 – 2025.	Pag. 35
Figura 1.4	Tipologia problemelor propuse pentru rezolvare participanților la Olimpiada Națională, clasele X - XII, în perioada 2019 – 2025	Pag. 36
Figura 2.1	Schema generală a organizării procesului de instruire prin divizare în nivele și etape	Pag. 54
Figura 2.2	Componentele calculatorului Raspberry Pi: Cabluri pentru interconectare și conectare la monitor; tastatură; placă de sistem cu procesor și porturi montate, sistem de răcire, bloc de alimentare, memorii	Pag. 56
Figura 2.3	Repartizarea conținuturilor pe domenii la etapa de motivare inițială	Pag. 58
Figura 2.4	Conținuturile teoretice ale etapei instrumentale A2 - limbaje de programare	Pag. 59
Figura 2.5	Structuri de date și clasificarea acestora după diverse criterii în funcție de domeniul de aplicare	Pag. 62
Figura 2.6	Principalele domenii de selecție a tematicii problemelor celebre propuse pentru rezolvare pe parcursul instruirii	Pag. 69
Figura 2.7	Procesul de rezolvare a problemelor din diverse domenii prin construcția modelelor matematice ale acestora	Pag. 74
Figura 2.8	Ilustrație la enunțul problemei Coroziunea metalului	Pag. 76
Figura 2.9	Modelul apropierei lichidelor la absorbția plăcilor extreme	Pag. 77
Figura 2.10	Modificarea modelului matematic și a algoritmului de rezolvare în funcție de subproblema rezolvată A – calcularea ariei unui cerc, B – calcularea ariei unui segment de cerc și a ariei unui poligon simplu, C – calcularea ariei unui poligon simplu și a multiplelor segmente de cerc	Pag. 78
Figura 2.11	Tipuri de complexitate și complexitatea structurilor de date	Pag. 80
Figura 2.12	Mulțimile vârfurilor $\{1, 3, 7\}$ $\{2, 8, 7, 4\}$ $\{4, 6\}$ – mulțimi independente. Mulțimile $\{2, 8, 7, 4\}$, $\{2, 4, 6\}$ sunt maximal independente, iar $\{1, 3\}$, $\{2, 4\}$ – nu	Pag. 83
Figura 2.13	Puncte de reper de studiere a tehnicilor de programare	Pag. 85

Figura 2.14	Ilustrarea principiilor de rezolvare a problemelor prin tehnica Divide et Impera	Pag. 88
Figura 2.15	Structuri de date complexe	Pag. 89
Figura 2.16	Legătura între heap și tabloul liniar cu n elemente	Pag. 91
Figura 2.17	Algoritmi semnificativi	Pag. 92
Figura 2.18	Modelul 3D. Conținuturile teoretice, Resurse pentru activități practice, colecții de probleme	Pag. 97
Figura 2.19	Profilul elevului performer fa Informatică	Pag. 107
Figura 3.1.a	Fișa de observație a activității participanților Olimpiadei Internaționale la Informatică (fragment). Fișa conține date despre punctajele acumulate, repartizarea succesului în timp (pe durata competiției), instrumente de selecție a datelor pe probleme separate	Pag. 113
Figura 3.1.b	Fișa de observație a activității celor instruiți pe platforma de pregătire CodeForces. Fișa conține date despre punctajele acumulate în rundele de concurs și dinamica progresului în funcție de complexitatea problemelor	Pag. 113
Figura 3.2	Rezultatele calitative ale centrelor Instituționale de Performanță A, B, D, E la Olimpiada Națională de informatică, perioada 2017 - 2025	Pag. 116
Figura 3.3	Progresul individual al membrilor echipelor olimpice naționale la Informatică perioada 2018 – 2024. Pentru participanții care continuă a căror participări încep până în 2018 sunt incluse seriile complete de rezultate	Pag. 118
Figura 3.4	Ratingul mondial al universităților la care și-au continuat studiile absolvenții centrului de performanță A, olimpici internaționali la disciplina Informatica	Pag. 120
Figura 3.5	Preferințele de implicare, pe tipuri de activități, a absolvenților centrului de performanță A pentru în activitatea Consiliului Olimpic la disciplina Informatica	Pag. 121

LISTA ABREVIERILOR

USM	–	Universitatea de Stat din Moldova
UTM	–	Universitatea Tehnică a Moldovei
USPC	–	Universitatea Pedagogică de Stat Ion Creangă
UST	–	Universitatea de Stat din Tiraspol
IOI	–	Olimpiada Internațională la Informatică
BOI	–	Olimpiada la Informatică a țărilor Balcanice
CEOI	–	Olimpiada la Informatică a țărilor Europei Centrale
JBOI	–	Olimpiada la Informatică a țărilor Balcanice pentru juniori
EJOI	–	Olimpiada Europeană la Informatică pentru juniori
EGOI	–	Olimpiada Europeană la Informatică pentru fete
DFS	–	Parcurgere în adâncime
BFS	–	Parcurgere în lățime
DP	–	Programare dinamică
MMI	–	Mulțime maximal independentă

INTRODUCERE

Actualitatea și importanța temei

Sfârșitul mileniului doi a marcat începutul transformărilor calitative în sistemul educațional mondial. Marcate de declararea începutului *erei informaționale* și *societății cunoașterii* aceste transformări vizează *tehnologizarea* instruirii, modificarea *paradigmei educaționale* și a celei *informaționale*, a modelului de *comunicare didactică*, dar și apariția unor noi științe, discipline școlare și cursuri care vin în susținerea efortului sistemic de creare a cetățeanului secolului douăzeci și unu – inteligent, capabil să învețe continuu, să se adapteze rapid la modificări majore ale mediului profesional, social și tehnologic-digital. Semnalată inițial în lucrările cercetătorilor din a doua jumătate a secolului XX, transformarea sistemului educațional începe să se manifeste vizibil doar după apariția premiselor tehnologice moderne în domeniile prelucrării informațiilor și a comunicațiilor.

Societatea informațională este caracterizată prin creșterea exponențială a cantității de informație, accelerarea ciclului de viață a informației, sporirea gradului de acces al membrilor societății la informațiile recente. Toate acestea modifică natura procesului de transformare a informațiilor în cunoștințe, sau, pe scurt – natura învățării. Schemele clasice ale proceselor educaționale sunt influențate pe de o parte de noile instrumente și metode tehnologizate, iar pe de altă parte – de modificarea priorităților în domeniile de instruire, dar și de creșterea majoră a competitivității, în special în zona științelor aplicate.

Actualizarea continuă a modelelor educaționale impune formarea și dezvoltarea competenței digitale pe tot parcursul vieții individului uman. În consecință „noua educație urmează să învețe individul cum să clasifice și să reclasifice informația, cum să evalueze veridicitatea ei, cum să modifice după necesitate categoriile, cum să treacă de la concret la abstract și invers, cum să analizeze problemele într-un nou aspect – cum să se instruiască singur. Analfabetul viitorului nu va fi persoana care nu poate citi – aceasta va fi persoana care nu a fost învățată să învețe” [1, p. 196]¹. De aici – o nouă sarcină în procesul educațional – formarea la elevi a gândirii critice, adaptive.

Informatica apare în calitate de știință teoretică în anii treizeci ai secolului trecut, pornind de la cercetările matematicienilor David Hilbert (*Entscheidungsproblem*, 1928), Kurt Gödel (*Über formal unentscheidbare Satze der Principia Mathematica und verwandter Systeme*; 1931), Alan Turing (*"On Computable Numbers, with an Application to the Entscheidungsproblem"*, 1936). În perioada celui de al doilea Război Mondial se dezvoltă rapid atât fundamentele teoretice, cât și direcțiile practice, aplicative. Începând cu anii cincizeci și până în anii nouăzeci ai secolului trecut,

¹ Herbert Gerjuoy, citat de Alvin Toffler, în „Future shock”.

paralel cu noțiunea de informatică este utilizată și noțiunea de *cibernetică*, ca știință despre dirijarea sistemelor automate [2, p. 56].

Ramurile principale ale Informaticii sunt *Informatica Teoretică* care ține de fundamentele matematice ale informaticii, *Informatica Aplicată*, care se ocupă de dezvoltarea limbajelor și mediilor de programare, a aplicațiilor și sistemelor software și *Informatica Tehnică*, care se ocupă de dezvoltarea sistemelor tehnice programabile.

Aspectul aplicativ al Informaticii impune în scurt timp apariția laboratoarelor de cercetare: MIT – Laboratorul de Sisteme Informaționale și Decizionale (1939), Laboratorul de Științe ale Calculatoarelor (1963) [3, p. 8]; Cambridge – Laboratorul de Calcule automate (1937) [4, p. 34] [5, p. 117]. În intervalul de ani 1945 - 1955 apar primele cursuri universitare de Informatică (Computer Science), iar mai târziu, odată cu apariția calculatoarelor personale, Informatica se extinde și ca disciplină de studiu în învățământul preuniversitar.

În instituțiile de învățământ preuniversitar Informatica apare în calitate de disciplină de studii în perioada anilor 1980 – 1985, fiind inclusă în trunchiul disciplinelor exacte. În consecință, disciplinei Informatica i-au fost asociate modelele pedagogice, strategiile de instruire și metodele de formare acceptate în domeniul științelor exacte.

Un element de primă importanță în dezvoltarea Informaticii în calitate de disciplină de studii devin activitățile pentru obținerea performanței – concursurile și olimpiadele școlare. Acestea formează un sistem complet începând cu anul 1987, când la cea de a 24-a conferință UNESCO din Paris, a fost acceptată propunerea matematicianul bulgar Blagovest Sendov de a organiza Olimpiadele Internaționale la Informatică, similare celor pentru celelalte științe exacte: matematică, fizică, biologie și chimie [6, p. 53]. Prima Olimpiadă Internațională la Informatică este organizată doi ani mai târziu în 1989, în Bulgaria.

În Republica Moldova Informatica în calitate de disciplină de studii în școlile medii de cultură generală este studiată începând cu anul 1985, iar prima ediție a Olimpiadei Republicane la Informatică se desfășoară în anul 1987 [7, p. 255]. Prima participare a echipei naționale a Republicii Moldova la Olimpiada Internațională de Informatică este atestată în anul 1996 [7, p. 256]

Specificul perioadei actuale este determinat de dezvoltarea rapidă a tehnologiilor digitale, dezvoltare care impune și dezvoltarea paralelă a fundamentelor teoretice, iar în consecință creează un deficit vizibil de cercetători și profesioniști în atât în domeniul Informaticii Aplicate cât și a Informaticii Teoretice. Compartimentele aplicative ale Informaticii se propagă tot mai intens pe spațiul de cunoaștere tradițional atribuit celorlalte științe exacte. Se dezvoltă domenii de cercetare mixte. Aceasta impune identificarea și dezvoltarea legăturilor fundamentale ale informaticii cu celelalte științe exacte, modernizarea permanentă a instruirii informatice, sporirea atractivității

activităților profesionale și de cercetare în domeniu pentru tinerii cetățeni. În particular, este impusă o modernizare continuă a compartimentelor și conținuturilor disciplinei școlare Informatica, menținerea și creșterea performanțelor elevilor la nivel național și internațional. De aici rezultă și tema cercetării: *aspecte interdisciplinare în instruirea de performanță la Informatică*.

Actualitatea temei este determinată de o serie de factori de diversă natură:

Sociali. Societatea modernă este una înalt tehnologizată, Informatica penetrează masiv toate domeniile de activitate umană. Competența digitală a devenit absolut necesară pentru existența și dezvoltarea umană în mediul digital al "obiectelor inteligente" [8, p. 7] [9, pp. 2 - 8] .

Economici. Domeniul TIC este unul înalt profitabil și poate contribui esențial la dezvoltarea economică a țării. Extinderea cantitativă și calitativă a acestei ramuri a economiei naționale presupune atragerea unui număr mare de profesioniști tineri, competenți, capabili să gândească critic și să ia decizii într-o manieră dinamică, folosind multiple criterii.

Educaționali. Formarea *generației nativ digitale* este imposibilă fără o verticală de instruire informatică calitativă, axată pe performanță, care să înceapă în același timp cu primele activități de învățare tradițională și să continue pe tot parcursul vieții. Formarea unor potențiali *profesioniști* în domeniul informaticii și tehnologiilor informaționale este imposibilă fără o instruire informatică de performanță, iar instruirea de performanță este imposibilă fără abordări moderne, conexiuni interdisciplinare, modele și strategii pedagogice multidimensionale [10, p. 63].

Importanța factorilor menționați pentru dezvoltarea strategică a țării este reflectată în o serie întreagă de acte de politici naționale: Strategia națională de edificare a societății informaționale "Moldova electronică" [11]; Strategia națională de dezvoltare „Moldova 2030” [12], Strategia de transformare digitală 2023 – 2030 [13], Programul național de specializare inteligentă a Republicii Moldova pentru anii 2024 – 2027 „Smart Moldova” [14], Codul Educației al Republicii Moldova [15].

Instruirea de performanță în domeniul Informaticii devine în acest context un element important pentru realizarea proiectelor naționale de dezvoltare socială și economică, dar și de promovare a imaginii țării pe plan internațional.

Efortul realizat pe parcursul anilor de comunitatea informatică a Republicii Moldova este unul semnificativ: pornind de la noțiuni și concepte informatice elementare în 1985 în doar 30 de ani s-a ajuns la un curriculum modern, axat pe formarea de competențe informatice, flexibil și în continuă dezvoltare, care menține rolul fundamentelor teoretice și a gândirii critice, combinându-le organic cu cele mai noi realizări din ramura aplicativă a Informaticii. În timp dezvoltarea și modernizarea Informaticii în calitate de disciplină școlară a fost posibilă datorită eforturilor

cercetătorilor Anatol Gremalschi, Ilie Lupu, Valeriu Cabac, Gheorghe Bostan, Liubomir Chiriac, și a multor alora, care au creat baza metodologică și de resurse pentru instruirea la această disciplină.

Curriculumul național la Informatică este doar un instrument, un set de indiciatori a direcțiilor de studiu, cu recomandări și exemple, care lasă profesorului suficiente nivele de libertate pentru utilizarea creativă a metodelor educaționale moderne. Prezența în sala de clasă, pe lângă profesor și elevi, a unui dispozitiv inteligent cum este calculatorul modern, conectat la multiple resurse, care poate răspunde practic la orice întrebare, prelucrând instant volume de informație pe care un om, oricât de inteligent ar fi, nu le poate asimila pe parcursul unei vieți, poate modifica radical procesul educațional. Pe lângă multe alte posibilități, utilizarea eficientă a accesului la informațiile specializate poate reduce din abstractizarea Informaticii, indicând explicit legăturile cu diverse domenii ale științei și vieții cotidiene. Altfel vorbind, instrumentele digitale pot umaniza Informatica, dezvăluind legăturile ei puternice cu natura și societatea, oferind studii de caz, posibilități de elaborare a proiectelor interdisciplinare, analize de bune practici de utilizare a Informaticii pentru Ecologie, Energetică, și desigur pentru Informatică în sine, în contextul performanței informatice, fără care ar lipsi progresul tehnologic și social din ultimele decenii. Probabil este timpul să se revadă modelele de interacțiune didactică, centrate pe elev: comunicarea didactică, mai puțin cea pentru instruirea de performanță, se va realiza în scurt timp nu între doi actori – profesor și elev – ci între trei, unul dintre care va fi un asistent inteligent digital.

Mai puțin cea pentru instruirea de performanță – pentru că în acest context educațional nu funcționează statisticile tradiționale, care pun în mișcare asistenții digitali. Situațiile și problemele pentru care se caută soluții nu sunt nici triviale, nici cotidiene, nici evidente. Dar ele pot deveni mai simple, mai clare, mai abordabile dacă sunt analizate prin prisma unei alte științe, nu neapărat apropiate Informaticii, folosind observații, bazate pe cunoașterea unor date, fapte, fenomene, care permit construirea modelelor mai exacte a acestor situații și probleme.

Problema cercetării

Problema cercetării este determinată de necesitatea elaborării unor modele noi de instruire de performanță la Informatică, care să extindă competențele elevilor în domeniile interdisciplinare: informatică – matematică, informatică – fizică, informatică – lingvistică, informatică – științe sociale, domenii deosebit de importante pentru viitorii cercetători și ingineri.

Informatica este o știință tânără, care se află în perioada dezvoltării rapide. Modelele, metodele de cercetare și paradigma de instruire sunt în continuă schimbare. Atât compartimentele teoretice, cât și cele aplicative mențin ritmul de dezvoltare, în mare măsură datorită unei

interacțiuni active cu celelalte științe exacte. Schemele de interacțiune se modifică permanent, devenind din ce în ce mai complexe și variate.

Astfel, apare și se extinde o contradicție între ritmul înalt de dezvoltare a științei și tehnologiei informatice și progresul relativ lent a produselor curriculare la disciplina școlară Informatică. Acest decalaj are tendința de creștere odată cu trecerea timpului, iar metode eficiente pentru diminuarea lui încă nu au fost identificate la nivel de sistem de învățământ național.

În contextul competițiilor informatice de rang internațional această contradicție se extinde și datorită diferențelor structurale majore între syllabus-ul internațional de performanță și curriculumul național la disciplina Informatică.

În contextul celor expuse, problema cercetării se formulează astfel în modul următor:

Care sunt demersurile teoretice și metodologice de instruire de performanță a elevilor la disciplina școlară informatică, pentru rezultate sustenabile la concursurile de informatică de nivel național și internațional.

Scopul cercetării: fundamentarea teoretică și metodologică a instruirii de performanță a elevilor la disciplina Informatica prin abordări interdisciplinare.

Ipoteza cercetării: instruirea elevilor la Informatică din perspectiva competițională va fi eficientă dacă:

- Se va baza pe principiul interdisciplinarității;
- Vor fi valorificate valențele formative a Informaticii din perspectiva interdisciplinarității;
- Va fi conceptualizat, validat și promovat modelul interdisciplinar de instruire de performanță a elevilor la disciplina Informatica.

Obiectivele cercetării

1. Stabilirea valențelor formative a informaticii ca știință și disciplină școlară din perspectiva instruirii de performanță informatică a elevilor;
2. Analiza experienței internaționale în domeniu: cercetarea modelelor de instruire în țările cu tradiții informatice din regiune și a liderilor internaționali;
3. Fundamentarea unui concept de instruire de performanță a elevilor la Informatică din perspectiva competițională;
4. Elaborarea modelului interdisciplinar privind instruirea de performanță a elevilor la disciplina Informatica;
5. Modelarea profilului elevului performer la Informatică;
6. Validarea modelului interdisciplinar privind instruirea de performanță a elevilor la disciplina Informatica.

Metodele de cercetare:

- Documentarea științifică
- Analiza și sinteza teoretică
- Generalizarea și sistematizarea abordărilor privind instruirea de performanță la disciplina Informatică
- Modelarea teoretică
- Analiza surselor de date valide
- Prelucrarea statistică – matematică a datelor.

Noutatea și originalitatea științifică este obiectivată de:

- Valorificarea valențelor formative a Informaticii ca știință și ca disciplină școlară din perspectiva instruirii de performanță a elevilor;
- Abordarea interdisciplinară a instruirii de performanță a elevilor la Informatică axată pe principii de integrabilitate a diferitor discipline școlare, în cazul dat: matematica, fizica, chimia, biologia, științele umanistice;
- Dezvoltarea competenței elevilor în Informatica competitivă prin complementarea cu noi concepte și operații (unități de competențe);
- Modelul de instruire performantă a elevilor la Informatică din perspectiva interdisciplinarității, cadrului formativ al disciplinei Informatica, profilului elevului performer la Informatică;
- Profilul elevului performer la Informatică din perspectiva cognitiv, motivațională, afectivă, comportamentală și socială.

Rezultatele obținute în cercetare: care au contribuit la soluționarea problemei științifice rezidă în fundamentarea teoretică și metodologică a instruirii de performanță a elevilor la Informatică din perspectiva competițională la nivel național și internațional.

Semnificația teoretică a cercetării constă în:

1. Dezvoltarea teoriei și metodologiei de predare-învățare a Informaticii prin conceptualizarea instruirii de performanță a elevilor din perspectiva competițională;
2. Descoperirea de noi valențe și oportunități ale Informaticii ca știință și disciplină școlară în dezvoltarea teoriei și practicii de instruire în genere și a instruirii de performanță în speță;
3. Argumentarea valorificării principiului de integrare în structura modelului interdisciplinar de instruire pentru performanță a elevilor la Informatică;
4. Conceptualizarea instruirii de performanță a elevilor la Informatică ca reper pentru elaborarea modelului interdisciplinar de instruire a elevilor din perspectiva competițională;

5. Construcția profilului elevului performer la Informatică ca țintă de atins în cadrul instruirii de performanță la disciplina Informatica.

Valoarea aplicativă a lucrării: modelul scalabil propus pentru instruirea elevilor capabili de performanțe înalte, poate fi aplicat atât în cadrul instruirii tradiționale, cât și în cadrul pregătirii de performanță. Creșterea eficienței învățării se atestă atât în cazul implementării infuzionale, cât și a celei modulare, impactul fiind mai pronunțat în condițiile racordării modelului propus la necesitățile formabililor.

Implementarea rezultatelor științifice s-a realizat în centrul instituțional de instruire pentru performanță în Informatică a rețelei de licee Orizont și în cadrul olimpiadelor de nivel național și internațional.

Aprobarea rezultatelor: rezultatele cercetării au fost prezentate în cadrul conferințelor republicane a profesorilor de Informatică, la ședințele catedrei Informatică și Tehnologii Informaționale (UST, mai recent - UPSC), precum și în cadrul conferințelor internaționale:

- Conferința Științifică Internațională „Abordări Inter/Transdisciplinare în Predarea Științelor Reale, (Concept Steam)”, Chișinău, 2024, noiembrie 01 – 02.
- Conferința științifică internațională asociată OII, ediția X, Kazan, Rusia. 2016, august 18 - 20.
- Conferința științifică națională „85 ani ai Învățământului Superior în Moldova”, 2015, Septembrie, 24
- Conferința științifică internațională „Educația de calitate în științe Exacte și ale Naturii. Obiective. Strategii. Perspective”. 2014, septembrie, 25-28.
- Conferința științifică internațională asociată OII, ediția VIII, Taipei, Taiwan. 2014, iulie 13 -20.
- Conferința științifică națională pentru învățământ Virtual. Brașov, Romania. 2012, noiembrie.
- Conferința științifică internațională "Tehnologii avansate de instruire - ALTA 2011", Kaunas, Lituania. 2011, noiembrie.

Lista publicațiilor:

1. Monografii (recomandate spre editare de consiliul științific/senatul instituției acreditate la profilul respectiv)

1.2. monografii colective - 2

1. **CORLAT, Sergiu, KARLSSON, Goran, BRAICOV, Andrei, STAH, Diana, HELLSTROM, Margaret.** Metodologia utilizării TIC în învățământul superior. F.E.P. Tipografia Centrală, 2011. 204 p. ISBN 978-9975-76-070-6.

2. CHIRIAC, Liubomir. (coord.) Evaluarea procesului de studiere a științelor reale și ale naturii din perspectiva inter/transdisciplinarității (concept STEAM). Chișinău: Tipografia Centrală, 2020. 252 p. ISBN 978-9975-117-50-0.

2. Articole în reviste științifice

2.1. în reviste din bazele de date Web of Science și SCOPUS - 1

1. GREMALACHI, Anatol, PRISĂCARU, Angela, **CORLAT, Sergiu**. Olympiads in informatics in Republic of Moldova. In: *Olympiads in Informatics*. Vol. 10, 2016. Ediția a 28-a, 12-19 august 2016, Kazan. Vilnius, Lituania: Vilnius University, 2016, pp. 255-262. ISSN 1822-7732

2.3. în reviste din Registrul National al revistelor de profil, cu indicarea categoriei

2.3.1. Articole în reviste de cat. B - 5

1. **CORLAT, Sergiu**. Direcții prioritare în dezvoltarea resurselor digitale pentru educație. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2020, nr. 2-3(120-121), pp. 4-7. ISSN 1810-6455.
2. **CORLAT, Sergiu**, MEHRYAR-RAD, Roxana. Metodologia utilizării parcurgerii în adâncime (DFS) la rezolvarea problemelor informatice de concurs. In: *Acta et commentationes. Științe ale Educației*. 2021, nr. 4(26), pp. 109-118. ISSN 1857-0623.
3. **CORLAT, Sergiu**, GUZUN, Veaceslav, VORONA, Victor. Finding N bits using $O(\frac{N}{\log N})$ sums. In: *Acta et commentationes. Științe Exacte și ale Naturii*. 2022, nr. 2(14), pp. 101-105. ISSN 2537-6284.
4. GLOBALA, Angela, **CORLAT, Sergiu**, GASNAȘ, Ala. Opțiuni de continuare a studiilor pentru elevii olimpici din Republica Moldova la disciplina informatica. In: *Acta et commentationes. Științe ale Educației*. 2020, nr. 3(21), pp. 23-31. ISSN 1857-0623.
5. GLOBALA, Angela, GASNAȘ, Ala, **CORLAT, Sergiu**. Inter și transdisciplinaritatea dintre disciplinele reale și ale naturii, abordate în conținuturile de informatică din sistemul preuniversitar. In: *Acta et commentationes. Științe ale Educației*. 2021, nr. 1(23), pp. 36-51. ISSN 1857-0623.

2.3.2. Articole în reviste de cat. C - 5

1. **CORLAT, Sergiu**. Modernizarea instruirii asistate de calculator din învățământul secundar. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2013, nr. 3(79), pp. 37-39. ISSN 1810-6455.
2. **CORLAT, Sergiu**. Modelul proiectiv de instruire pentru utilizarea TIC în învățământul superior. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2013, nr. 4(80), pp. 5-9. ISSN 1810-6455.

3. GREMALSCHI, Anatol, **CORLAT, Sergiu**. Rolul interdisciplinarității Matematică-Informatică în pregătirea elevilor de performanță. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2011, nr. 4(68), pp. 46-49. ISSN 1810-6455.
4. GREMALSCHI, Anatol, **CORLAT, Sergiu**. Instruirea de performanță la Informatică în contextul curriculumului general . In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2011, nr. 5-6(69-70), pp. 74-78. ISSN 1810-6455.
5. **CORLAT, Sergiu**. Curriculumul Național – 2019 la Informatică: modernizare structurală și de conținut. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2019, nr. 4-5(116-117), pp. 53-55. ISSN 1810-6455.

2.3.3. Articole în alte reviste din RM - 8

1. **CORLAT, Sergiu**. Integrarea portofoliilor de predare și învățare. Metaportofoliul. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2010, nr. 6(64), pp. 8-12. ISSN 1810-6455.
2. **CORLAT, Sergiu**. Elemente de instruire la distanță și de autoinstruire în pregătirea elevilor pentru concursurile informatice Instruirea la distanță. In: *Revista Didactica Pro..., revistă de teorie și practică educațională*. 2010, nr. 6(64), pp. 29-31. ISSN 1810-6455.
3. LUPU, Ilie, GREMALSCHI, Anatol, **CORLAT, Sergiu**. Proiectarea curriculară în învățământul de performanță la informatică . In: *Acta et commentationes. Științe ale Educației*. 2014, nr. 2(5), pp. 55-61. ISSN 1857-0623.
4. LUPU, Ilie, **CORLAT, Sergiu**. Spațiul informațional educațional – o nouă paradigmă a autoinstruirii în mediul digital. In: *Acta et commentationes. Științe ale Educației*. 2013, nr. 2(3), pp. 138-145. ISSN 1857-0623.
5. **CORLAT, Sergiu**. Aspecte generale de instruire la distanță în învățământul superior. In: *Acta et commentationes. Științe ale Educației*. 2012, nr. 1, pp. 138-144. ISSN 1857-0623.
6. GLOBA, Angela, **CORLAT, Sergiu**. Didactic aspects regarding to creating test sets for competition problems. In: *Acta et commentationes. Științe ale Educației*. 2019, nr. 4(18), pp. 73-85. ISSN 1857-0623.
7. **CORLAT, Sergiu**, IVANOV, Lilia. Problems of junior contestants training for participation in national and international programming competitions. In: *Acta et commentationes. Științe ale Educației*. 2019, nr. 4(18), pp. 120-124. ISSN 1857-0623.
8. ESANU, Mihai, **CORLAT, Sergiu**. A comparative study of maximum matching algorithms. In: *Acta et commentationes. Științe ale Educației*. 2024, nr. 1(35), pp. 97-102. ISSN 1857-0623.

3. Articole în culegeri științifice

3.2. în lucrările conferințelor științifice internaționale (Republica Moldova) - 1

1. BRAICOV, Andrei, CORLAT, Sergiu. Un model de activitate educațională STEM implementată prin metoda instruirii în bază de probleme. In: *Abordări inter/transdisciplinare în predarea științelor reale, (concept STEAM)*. Ediția 3, 27-28 octombrie 2023, Chișinău. Chișinău: CEP UPSC, 2023, pp. 228-235. ISBN 978-9975-46-813-8

3.4. în lucrările conferințelor științifice naționale din RM - 5

1. LUPU, Ilie, GREMALSCHI, Anatol, **CORLAT, Sergiu**. Implementarea manualelor digitale în învățământul din Republica Moldova. In: *Învățământul superior din Republica Moldova la 85 de ani Probleme actuale ale didacticii științelor*. Vol. 3, 24-25 septembrie 2015, Chișinău. Chișinău, Republica Moldova: Universitatea de Stat din Tiraspol, 2015, pp. 141-149. ISBN 978-9975-76-161-1
2. BORDAN, Valeriu, BRAICOV, Andrei, **CORLAT, Sergiu**, LUPU, Ilie, SALI, Larisa, TE-LEUCĂ, Marcel. Experiențe de formare continuă a profesorilor preuniversitari de matematică și informatică: studiu de caz. In: *Învățământul superior din Republica Moldova la 85 de ani Probleme actuale ale didacticii științelor*. Vol. 3, 24-25 septembrie 2015, Chișinău. Chișinău, Republica Moldova: Universitatea de Stat din Tiraspol, 2015, pp. 201-209. ISBN 978-9975-76-161-1
3. **CORLAT, Sergiu**, IVANOVA, Lilia. Clasificarea structurală a manualelor digitale. In: *Școala modernă: provocări și oportunități*. 5-7 noiembrie 2015, Chișinău. Chișinău, Republica Moldova: Institutul de Științe ale Educației, 2015, pp. 111-114. ISBN 978-9975-48-100-7.
4. BRAICOV, Andrei, **CORLAT, Sergiu**, VEVERIȚĂ, Tatiana. Despre competența digitală a cadrelor didactice din învățământul profesional tehnic. In: *Învățământ superior: tradiții, valori, perspective Științe Exacte și ale Naturii și Didactica Științelor Exacte și ale Naturii*. Vol. 1, 1-2 octombrie 2021, Chișinău. Chișinău, Republica Moldova: Universitatea de Stat din Tiraspol, 2021, pp. 37-41. ISBN 978-9975- 76-360-8.
5. **CORLAT, Sergiu**, IVANOVA, Lilia. Aplicații Social Software pentru elevi. Tehnici avansate de căutare a informației pe WEB. In: *Educația pentru dezvoltare durabilă: inovație, competitivitate, eficiență*. 18-19 octombrie 2013, Chișinău. Chișinău: Institutul de Științe ale Educației, 2013, pp. 608-612. ISBN 978-9975-48-056-7

8. Lucrări științifico-metodice și didactice - 2

8.1. manuale pentru învățământul preuniversitar (aprobate de ministerul de resort)

1. GREMALSCHI, Anatol, **CORLAT, Sergiu**, BRAICOV, Andrei. Informatică. Manual pentru clasa XII. Editura „Știința”, 2015. 144. p. ISBN 978-9975-67-984-8.

2. GREMALACHI, Anatol, **CORLAT, Sergiu**, BRAICOV, Andrei. Informatică. Manual pentru clasa XII. Editura „Știința”, 2025. 200 p. ISBN 978-9975-85-492-4.

8.2. manuale pentru învățământul universitar (aprobate de senatul instituției) - 5

1. **CORLAT, Sergiu**, GREMALACHI, Anatol. Metodologia rezolvării problemelor de geometrie computațională. Universitatea de Stat din Tiraspol, 2013. 100 p.
2. **CORLAT, Sergiu**, GREMALACHI, Anatol. Grafuri. Metodologia predării în cadrul instruirii de performanță la disciplinele Matematică & Informatică. Universitatea de Stat din Tiraspol, 2014. 158 p. ISBN 978-9975-76-122-2.
3. **CORLAT, Sergiu**, GREMALACHI, Anatol. Grafuri. Noțiuni, algoritmi, implementări. Universitatea Academiei de Științe din Moldova, 2012. 124 p.
4. **CORLAT, Sergiu**, CORLAT, Andrei. Calcul diferențial și Integral. Culegere de probleme. Universitatea Academiei de Științe din Moldova, 2012. 66 p.
5. **CORLAT, Sergiu**. Algoritmi și probleme de geometrie computațională. Editura „Prut Internațional”, 2009. 72 p. ISBN 978-9975-69-377-6.

8.3. alte lucrări științifico-metodice și didactice - 7

1. **CORLAT, Sergiu**, IVANOV, Lilia, BÂRSAN, Valentin. Informatică. Ghid de implementare a curriculumului modernizat pentru treapta liceală. Editura „Cartier” 2010. 112 p. ISBN 978-9975-79-647-7.
2. **CORLAT, Sergiu**, IVANOV, Lilia. Calculatorul în predare / învățare. Editura „Știința”, 2007. 52 p. ISBN 978-9975-67-265-8.
3. **CORLAT, Sergiu**. Caractere digitale. Catalog. Editura USM, 2005.
4. **CORLAT, Sergiu**, IVANOV, Lilia. Calcul numeric. Curs de lecții la Informatică. CCRE „Presa”, 2004. 96 p.
5. **CORLAT, Sergiu**. Sistemul editorial PageMaker. Editura USM, 2002.
6. **CORLAT, Sergiu**. Ediții electronice în format HTML. Editura USM, 2002. 68 p. ISBN 9975-70-110-8

Cuvinte cheie: *disciplina Informatica, curriculum național, olimpiade școlare, performanță informatică, instruire de performanță, model de instruire.*

Sumarul compartimentelor tezei

Teza este alcătuită din Introducere, trei capitole, Concluzii și recomandări finale, conține o listă bibliografică de 202 ediții tipărite, electronice, site-uri și platforme web.

Introducerea conține descrierea problemei de cercetare, scopului și a obiectivelor cercetării, rezultatele cercetării și noutatea lor științifică, importanța teoretică a rezultatelor și modul de validare a acestora.

Capitolul 1 conține o scurtă descriere a devenirii Informaticii ca știință exactă și dezvoltarea domeniilor ei aplicative, disciplină de studii și teren de competiții intelectuale. Sunt punctate principalele etape istorice și personalități ale științei de importanță majoră, care au dus la formularea unor probleme sau domenii de cercetare fundamentale. Se studiază modelele de instruire informatică în universități și în învățământul preuniversitar. Este analizat și descris modelul de organizare a competițiilor informatice în țară și descrise etapele organizării Olimpiadei Naționale la Informatică, inclusiv procesul selecției lotului olimpic la această disciplină. Tot aici este realizată o scurtă analiză a tipologiei subiectelor propuse în cadrul Olimpiadei, sunt identificate cele mai frecvente categorii de probleme, folosite în concurs în dependență de vârsta participanților. Se constată o divergență tot mai accentuată în timp între Curriculumul Național la disciplina Informatică și prevederile Syllabus-ului Internațional pentru performanță, utilizat la organizarea Olimpiadelor Internaționale la Informatică. Sunt studiate și elementele interdisciplinare ale Informaticii, ca disciplină școlară, în special legătura cu științele exacte: fizica, matematica, chimia, biologia și modelele de integrare a problemelor din diferite domenii în conținuturile informatice. Capitolul finalizează cu o serie de concluzii specifice.

Capitolul 2 este dedicat soluției construite în timp pentru lichidarea divergențelor între curriculumul standard și cel internațional de performanță. Analiza experienței internaționale scoate în evidență prezența centrelor de performanță formalizate în învățământul clasic în forma liceelor cu profil interdisciplinar – matematic – informatic sau fizic – informatic. În lipsa unor asemenea profiluri în învățământul general din Republica Moldova, ele pot fi simulate, folosind posibilitățile oferite de sistemul educațional pentru activități neformale – cercuri, cluburi – orice structuri care presupun activități de instruire în afara orelor de curs. În baza analizei, dar și a propriei experiențe acumulate în calitate de mentor a fost realizat un model de instruire, axat pe succes, care are la bază trei principii fundamentale: *interdisciplinaritatea* – în sens de cunoaștere a principiilor fundamentale pentru toate științele exacte, ceea ce simplifică procesele de înțelegere a problemelor și crește eficiența în condiții de concurs; *educarea rezistenței la stres* prin participarea periodică la concursuri neoficiale și runde de antrenament de antrenament pe

platforme automate de evaluare, și *organizarea distribuită a comunicării* – cu comunicare asincronă la distanță pentru o monitorizare mai riguroasă a progresului fiecărui instruit.

Primul paragraf este dedicat analizei fenomenului de performanță în Informatica școlară, definirii rolurilor actorilor educaționali, identificării specificului instruirii, modernizării syllabus-ului de performanță național, cu includerea noilor compartimente, propuse între timp în Syllabus-ul internațional. Compartimentele noi vin din zona matematică și fizică, ceea ce confirmă corectitudinea modelului elaborat.

Descrierea modelului este integrată în paragraful doi. Ea este cât se poate de simplă – urmează principiul de funcționare a CRYPTEX-ului – rezultatul se obține doar dacă toate etapele au fost parcurse și setate în poziția corectă prin rezolvarea unui set exhaustiv de mare de probleme și exerciții semnificative din punct de vedere didactic, selectate pe parcursul anilor. Paragraful trei este dedicat analizei specificului activităților de cunoaștere, care însoțesc procesul de instruire în baza modelului descris. Ultimul paragraf – al patrulea, descrie profilul elevului performer – format în baza modelului de instruire elaborat de autor.

Capitolul conține descrierea conținuturilor fiecărei din cele nouă etape de formare, grupate în trei nivele de performanță, fiecare nivel având ținta sa competiții locale(1), competiții naționale(2) și internaționale(3). Formarea echilibrată a performanței se datorează interconectării etapelor în cadrul fiecărui nivel, iar stabilitatea modelului este obținută prin o „tijă centrală” – concursurile care în timp devin mai multe și mai complexe.

Capitolul trei analizează performanța elevilor instruiți cu modelul CRYPTEX în comparație cu toți ceilalți membri a Echipei Naționale la Informatică pentru IOI. Pentru observații au fost folosite fișele, generate de sistemele de evaluare ale competițiilor internaționale, dar și fișele de la competițiile naționale, ceea ce a permis analiza în paralel a creșterii performanței la concursurile naționale cu creșterea la concursurile internaționale. Statisticile, dar și concluziile de sfârșit de capitol demonstrează funcționalitatea și eficiența modelului elaborat. Mai mult, paragraful trei, dedicat analizei activităților de performanță după absolvirea liceului, demonstrează o tendință puternică a absolvenților de a rămâne în zona activităților competiționale de performanță.

Concluziile și recomandările generale formează ultima parte a tezei. Ele se referă la posibilitățile de extindere a modelului în diverse centre de performanță instituționale în țară, recenterarea pe alte discipline de studii (matematica, fizica), modernizarea periodică a conținuturilor și instrumentelor folosite, condiții și circumstanțe pentru continuarea cercetării.

1. INSTRUIREA DE PERFORMANȚĂ ÎN INFORMATICĂ: CADRUL TEORETIC

1.1. Informatica - știință exactă și disciplină școlară

Poziționată ca o știință independentă începând cu anii 30 ai secolului trecut, Informatica se sprijină pe rezultatele obținute pe parcursul timpurilor în mai multe științe: matematică, fizică, logică. Bazele matematice și logice ale informaticii cu certitudine se regăsesc în Asia, în descrierea detaliată a numerelor Hindu, efectuată de matematicianul Muhammad ibn Musa al-Khwarizmi. Modelul formal, propus de al-Khwarizmi a fost preluat de generațiile următoare pentru descrierea oricărei reguli specifice de procedură sau operațiune matematică, cum ar fi metoda euclidiană pentru determinarea celui mai mare divizor comun a două numere, primind și denumirea derivată din numele autorului - "algorism" sau "algoritm" [16, p. 206].

Componenta materială a Informaticii își are originile în cel mai simplu instrument pentru calculul automatizat – abacus – începutul utilizării căruia datează conform diferitor surse, în diferite regiuni ale lumii cu perioada între anii 2700 (Mesopotamia) [17, p. 11] și 1500 (Egipt) [18, p. 27] înaintea erei noastre. Dezvoltarea ulterioară este rezultatul descoperirilor medievale și post-medievale în domeniul mecanicii, care au permis crearea primelor dispozitive pentru calculul automat. Acestea au fost folosite în perioada sec. XVII – XX, începând cu mașinile pentru calcule aritmetice a lui Babbage, Pascal, etc. și finalizând cu dispozitivele „de buzunar”, produse în anii treizeci a secolului trecut de compania IBM.

Ulterior, dezvoltarea conceptelor dispozitivelor informatice a trecut din domeniul mecanicii în cel al fizicii curentului electric, semiconductoarelor și opticii. Rezultatul aplicativ al utilizării principiilor și proprietăților fundamentale a entităților care țin de aceste domenii au devenit sistemele electrice (ulterior electronice) de calcul, cunoscute sub denumirea generică de *sisteme informatice* sau *calculatoare*. Evoluția acestora a fost și se află în continuare în dependență directă de progresul științific în acele domenii a fizicii moderne, care permit codificarea eficientă a informației, transferul rapid al și procesarea în timp real a ceea ce înseamnă *big data*.

Componenta logică a Informaticii s-a dezvoltat ca o extensie a logicii matematice, a cărei baze au fost puse de matematicianul englez George Boole în lucrările sale *Mathematical Analysis of Logic* [19] și *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities* [20]. Or, tocmai componenta logică a permis dezvoltarea principiilor, iar ulterior a sistemelor eficiente de codificare a datelor, folosind coduri binare. Aceleași principii stau la baza proiectării dispozitivelor pentru calcule aritmetice și logice. Tot ele, ca parte a matematicilor aplicate, au permis elaborarea principiilor și dezvoltarea sistemelor de comprimare

a datelor – de exemplu arborii Huffman – fără care utilizarea resurselor multimedia ar deveni practic imposibilă.

Domeniul de control al sistemelor digitale, sau componenta cibernetică a informaticii este în strânsă legătură cu dezvoltarea compartimentelor deja clasice ale matematicii aplicate, cum ar fi teoria algoritmilor, structuri de date, geometrii computaționale, dar și a unor domenii interdisciplinare mai puțin obișnuite, de tip: *mulțimi fuzzy*, *algoritmi genetici*, *rețele neuronale*, care permit crearea unor modele eficiente de analiză a datelor mari, dar și dezvoltarea unor metode inovative pentru rezolvarea problemelor de complexitate sau dimensiune sporită.

Integrarea rezultatelor științifice din diverse domenii ale științelor exacte a dat naștere unui nou compartiment informatic – *mechatronica*, care reprezintă un nivel nou de dezvoltare a ciberneticii, fiind orientat către controlul sistemelor digitale care pot implementa conceptul de *acțiune fizică*, spre deosebire de sistemele informatice, orientate către producerea unui rezultat virtual, sau în formă de date digitale.

Predecesorul direct al Informaticii este matematica, or, problema realizării calculului automat vine de la numărul mare de calcule matematice complexe, necesar a fi efectuate pentru rezolvarea unor probleme ingineresti de proiectare – problemă, devenită extrem de acută la sfârșitul secolului XIX – începutul secolului XX. Printre primele modele teoretice se regăsesc cele propuse de Alan Turing, care reduce procesul de calcul la acțiunile unui sistem cu număr finit de stări. Acestea au fost la baza mașinii Turing, permițând să se facă și primele estimări privind complexitatea problemelor, capacității de calcul a sistemelor informatice dar și a limbajelor formale.

Fiind introdusă noțiunea de complexitate, a fost efectuată și clasificarea spațiului de probleme, în urma căreia au fost identificate clasele P, NP, NP-hard și s-a introdus și noțiunea de complexitate a algoritmilor. În 1961 Tony Hoare descrie unul dintre cei mai controversați algoritmi de sortare Quick Sort, [21] care deschide direcția de analiză a algoritmilor, dezvoltată ulterior cu multă perseverență de numeroși cercetători din domeniul matematicii aplicate și Informaticii, lucrările lui Donald Knuth [22], [23], [24], [25], [26], fiind cele mai relevante în acest sens.

Pe de altă parte, din compartimentele matematicii clasice se dezvoltă alte domenii aplicative, cum ar fi structurile de date și gestionarea eficientă a acestora. Teoria grafurilor, gândită de Euler ca un instrument de demonstrație matematică, se transformă într-o mulțime de operații de diferită complexitate, realizate asupra unei structuri de date abstracte – graf [27], [28]. În calitate de structură de date grafurile dau naștere la o mulțime de meta operații de natură algoritmică, implementate în majoritatea sistemelor inteligente digitale.

Tot grafurile au făcut posibile cercetările moderne în domeniul Inteligenței Artificiale. Analiza modelelor lingvistice este realizată pe rețele neuronale, care , de facto reprezintă o

categorie de grafuri cu proprietăți specifice [29]. Combinatorica, datorită multitudinii de operații specifice, devine o parte indispensabilă a Informaticii moderne – este suficient să fie enumerate asemenea compartimente ca: generarea funcțiilor, principiul de includere – excludere, Inversia Möbius, fluxurile în rețea, cunoscute astăzi oricărui cercetător în domeniul informaticii teoretice [30].

Geometria, una dintre primele ramuri ale matematicii, se transformă radical odată cu apariția serviciilor de localizare prin satelit, iar mai târziu - cu modelarea 3D și serviciile de telefonie mobilă. Problemele înfășurătorii convexe [31, pp. 114 - 181], a Poligonului și Diagramei Voronoi, [32], triangularizării arbitrare și triangularizării Delaney au căpătat aplicații reale odată cu apariția sistemelor informatice performante [32].

Transformările de coordonate, care însoțesc animația digitală sunt parte din algebra liniară. Tot aici, vectorii, în zona structurilor de date, vin să se suprapună pe vectorii tradiționali, cunoscuți în algebră ca fiind șiruri liniare de valori numerice cu diverse proprietăți. Toate realizările curente obținute de industria vizuală au la origini câteva transformări elementare ale matricelor, utilizarea inteligentă a operațiilor logice și extrem de multe calcule aritmetice, care nu pot fi realizate altfel, decât cu ajutorul sistemelor de calcul [33].

Aplicațiile Informaticii, ca știință s-au extins și asupra compartimentelor tradițional teoretice ale matematicii, cum ar fi Analiza Matematică, Calculul Diferențial și Integral. Metodele numerice de calculare a soluțiilor problemelor matematice complexe sunt bine studiate, iar câteva dintre ele, cum ar fi rezolvarea ecuațiilor algebrice și transcendente, rezolvarea sistemelor de ecuații liniare și calcularea integralei definite se studiază inclusiv în învățământul preuniversitar [34] [35].

Varietatea modurilor de utilizare a resurselor informatice este în continuă creștere. În consecință, rezultatele cercetărilor, obținute cu ajutorul instrumentelor informatice, devin „materia primă” pentru cercetările informatice ulterioare. Calculatorul cuantic nu ar fi fost posibil de realizat fără utilizarea sistemelor informatice clasice, acestea din urmă apărând doar datorită cercetărilor teoretice, realizate în sec. XVIII – XIX [17]. Mai mult, ar fi fost imposibilă dezvoltarea sistemelor moderne de Inteligență Artificială, care funcționează conform legilor matematicii discrete, teoriei grafurilor, teoriei probabilităților, statisticii, prelucrând volume mari de date cu ajutorul algoritmilor eficienți și a structurilor de date optimizate pentru efectuarea operațiilor asupra informației [36].

Dezvoltarea rapidă a informaticii ca știință ar fi fost imposibilă fără un număr suficient de mare de cercetători în domeniu. Formarea acestora era, și rămâne în continuare, misiunea universităților. Primele structuri organizaționale universitare dar și primele cursuri specializate

de Informatică în sensul de Computer Science au fost lansate în perioada 1946 – 1955 la mai multe universități, aproape simultan: 1950 – Universitatea Massachusetts, 1953 – Universitatea Cambridge, 1952 - Universitatea din Stanford [37]. În Republica Moldova (în acea perioadă Republica Sovietică Socialistă Moldovenească) la Universitatea de Stat a fost deschisă Catedra de Matematică Aplicată în anul 1961, iar câțiva ani mai târziu, tot la USM a fost deschis unul dintre cele mai performante centre de calcul universitare pentru acea perioadă [38].

Ca disciplină școlară Informatica este introdusă în școlile de cultură generală din Uniunea Sovietică în anii 1985 – 1987. În particular, în Republica Moldova Informatica este inclusă în programul școlar, începând cu anul 1985 [7]. Procesul de introducere a Informaticii pe scară largă în calitate de disciplină de studii în învățământul preuniversitar a fost precedat de diverse experiențe, printre care oferirea posibilităților de studiere a bazelor Informaticii în baza unor resurse publicate în revistele științifice, prin corespondență. Una dintre aceste oportunități a fost Școală Tânărului Programator, a cărei materiale erau publicate în revista științifică de matematică și fizică Kvant. Perioada de funcționare a acestei școli au fost anii 1979 – 1982 [39], [40]. Deși a fost o experiență de instruire la distanță, autorii au reușit să stabilească corect compartimentele inițiale ale cursului care urma să fie implementat în școli, precum și nivelul lor de complexitate [41].

Primul manual elaborat pentru școala generală deja conținea compartimente de programare și elaborare și implementare a algoritmilor [42], [43]. O trecere în revistă a cuprinsului acestui manual permite identificarea direcțiilor prioritare pentru Informatica școlară ca pentru o știință exactă, fundamentală, dar cu elemente aplicative interdisciplinare pronunțate:

- Algoritmi și limbaje algoritmice
- Elaborarea algoritmilor pentru rezolvare de probleme
- Arhitectura sistemelor informatice
- Limbaje de programare (Basic)
- Rolul sistemelor informatice în societatea modernă. Perspectivele de dezvoltare a sistemelor informatice.

După obținerea independenței, în Republica Moldova apar propriile manuale și culegeri de probleme la Informatică pentru învățământul preuniversitar. Printre edițiile care au educat primele generații ale tinerilor informaticieni din Moldova s-au remarcat: *Culegere de probleme de informatică* de Gheorghe Bostan [44], *Informatica Limbajul Pascal Manual pentru clasele IX – XI* de Anatol Gremalschi, Iurie Mocanu și Ion Spinei [45], *Totul despre calculatoare pentru cei mici, părinți și bunici*, de Sergiu Corlat și Boris Singer [46]. În perioada 1996 – 2003 apar și

manualele pentru clasele a șaptea și a opta, apoi manualele pentru clasele a noua, a zecea și a unsprezecea, separate pe ani de studii, conținând compartimentele de editoare de teste, tabele de calcul, baze de date, sisteme operaționale, arhitectură a calculatorului și rețele de calculatoare. La apariția lor au contribuit: Anatol Gremalschi, Ludmila Gremalschi, Grigore Vasilache, Ion Spinei, Iurie Mocanu. Ultimul gol în acoperirea curriculară cu resurse de învățare a fost acoperit în 2003, odată cu apariția volumului *Calcul numeric Curs de lecții* de Sergiu Corlat și Lilia Ivanov [47], iar ultimul manual, pentru clasa a douăsprezecea, scris de Anatol Gremalschi, Sergiu Corlat și Andrei Braicov, a apărut tocmai în 2010 [35].

În calitate de limbaj de programare, utilizat în instituțiile de învățământ din Republica Moldova se folosea limbajul Pascal, dezvoltat de Niklaus Wirth, cercetător elvețian în domeniul Informaticii teoretice, în perioada anilor 1968 – 1970 ca o extindere a limbajului de programare ALGOL X. Complet procedural, înzestrat cu toate instrumentele informatice didactice necesare – sistem de asistență; feedback complet pentru depanarea erorilor; mediu de programare comod și intuitiv organizat; număr redus de cuvinte-cheie – Pascal-ul era indiscutabil cel mai recomandat instrument în acea perioadă de timp pentru studierea bazelor programării în învățământul preuniversitar [48], [49].

În 2010 are loc prima și „aerisire” a curriculumului la Informatică, în urma căreia studiul mai multor tehnici de programare și structuri de date a fost transferat în zona opțională – aici s-au regăsit elementele de programare dinamică, căutare binară, algoritmi de sortare – toate de importanță majoră în pregătirea de performanță. Totuși, modulele obligatorii de programare în baza unui limbaj de programare de nivel înalt se regăsesc în curriculum începând cu clasa a noua, ceea ce păstrează speranța de participare a elevilor din echipele de juniori la competițiile internaționale în baza unor selecții naționale, la fel cum se întâmplă de exemplu, în România și Bulgaria. În următorii ani apar curriculum-uri specializate, se introduce programarea vizuală în Scratch, alte module de învățare, apropiate mai mult de câmpul tehnologic, decât de cel științific. După anul 2015 apar primele tentative de programare a roboților, apare o disciplină de studiu opțională nouă – robotica, cu tentă pronunțată algoritmică și de programare.

În redacția curriculumului național la Informatică din anul 2019 [50] au loc schimbări structurale vizibile față de curriculumul precedent, modulele de programare clasică devenind opționale în gimnaziu – *Primele mele programe* (clasa a șaptea); *Implementarea algoritmilor în medii text de programare* (clasa a opta); *Prelucrarea datelor structurate în medii text de programare* (clasa a noua); La fel, modulele complexe de programare din etapa de liceu [51], au fost grupate în clasa a douăsprezecea, atunci când cunoștințele obținute și abilitățile formate nu mai sunt atât de importante, datorită graficului de organizare a competițiilor internaționale.

Din curriculumul 2019 au dispărut recomandările privind limbajul de programare utilizat – astfel, în funcție de clasă, profesorul a căpătat posibilitatea să aleagă un limbaj de programare mai modern, mai flexibil, mai eficient. Mai multe cadre didactice deja optaseră la acea vreme pentru limbajele C/C++, au apărut și elevi care studiază Python. Totuși, evaluările naționale, care se desfășoară în scris, presupun alegerea opțiunii de susținere a examenului de bacalaureat la Informatică doar utilizând Pascal sau C/C++.

Prin urmare, curriculumul la Informatică, ediția anului 2019 a oferit posibilitatea să fie studiată programarea în instituțiile unde profesorii sunt suficient de bine pregătiți, a oferit o diversitate mai mare de instrumente informatice pentru studiu, dar a micșorat prin aceasta spațiul de selecție al elevilor pentru instruirea de performanță – numărul de copii care au acces formal la cunoștințele necesare pentru instruirea de performanță la Informatică, începând cu clasele mici [52].

În calitate de instrument compensatoriu a fost lansat proiectul *Tekwil în fiecare școală*. Scopul acestuia era de a interconecta profesorii din școli cu elevii în grupuri distribuite, oferind elevilor posibilitatea să comunice cu profesori de calitate, iar profesorilor – să ghideze elevi cu auto motivație puternică. Mai mult, au fost realizate și sesiuni de instruire pentru cadrele didactice, pe parcursul primilor doi ani de proiect. Modelul proiectului a fost elaborat în corespundere cu principiile internaționale pentru comunicare și instruire la distanță [53] [54] [55] [41]. Conținuturile – orientate către modernizarea instruirii informatice în învățământul preuniversitar. Cursurile oferite de proiect sunt modulare, independente de vârsta elevilor, și vizează *Programarea algoritmilor în limbajul C/C++* (nivel de bază și avansat), *Programarea aplicațiilor mobile*, *Designul Web*, *Inteligența artificială*.

Analiza comparativă a curriculumului național la informatică și a curriculumului la disciplina Informatica, după care se face instruirea în gimnaziile și liceele din România și Bulgaria stabilește un raport aproximativ egal, atât pentru orele de curs rezervate, cât și pentru conținuturile studiate. Dar, și România, și Bulgaria au dezvoltat rețele de licee (colegii naționale) cu profil informatic și matematic-informatic teoretic, cu plan-cadru specific, în care se studiază un număr mare de subiecte informatice, matematice, fizice și logice, absolut necesare pentru obținerea performanțelor informatice [56] [57]. Posibilitatea creării unor asemenea rețele de licee specializate este determinată nu doar de numărului populației acestor două țări: circa 22 milioane locuitori în România și 6,5 milioane în Bulgaria, dar și de existența unor bogate tradiții în școlile naționale matematice și informatice. Este suficient să se menționeze că prima Olimpiadă Internațională la Matematică a fost organizată în 1959 în România [58], iar prima Olimpiadă Internațională la Informatică – în anul 1989, Bulgaria, Pravetz [59].

În tabelul 1.1 sunt prezentate diferențele între modul de alocare a orelor de informatică și tehnologii informaționale în liceele teoretice generale și specializate din România, Bulgaria și Republica Moldova:

Tabelul 1.1. Repartizarea orelor de Informatică și Tehnologii Informaționale (săptămânal) în liceele teoretice și liceele cu profil matematic – informatic în România, Bulgaria și Republica Moldova

	România				Bulgaria				Republica Moldova	
	General		Mate-info		General		Mate-info		Umanist	Real
	TI	I	TI	I	TI	I	TI	I	1 (nu se aplică divizarea)	
Clasa IX	2	1	2	1	2	-	2	4	1	2
Clasa X	1	0	1	1	2	-	2	4	1	2
Clasa XI	2	0	4	4	-	-	-	4	1	2
Clasa XII	1	0	4	4	-	-	-	4	1	2
TI – tehnologii Informaționale, I - informatică										

O deosebire majoră în structura curriculară pentru domeniul informatic este separarea în țările vecine a modulelor informatice cu tentă tehnologică pronunțată, cum ar fi *editarea textelor*, *utilizarea foilor de calcul*, *grafica și designul digital*, într-o disciplină separată – tehnologia informației. Astfel, modulele în cadrul disciplinei Informatica sunt orientate la dezvoltarea competențelor informatice tradiționale: gândire critică, abstractizare, modelare, construcție de modele informatice.

În concluzie – Curriculumul național la Disciplina Informatică nu este suficient de extins pentru a acoperi toate (sau cel puțin majoritatea modulelor) necesare pentru a forma elevi, a căror performanță informatică este suficientă pentru obținerea rezultatelor în cadrul olimpiadelor regionale și internaționale la disciplină [60] [61]. Totuși, spre deosebire de țările vecine, în Republicii Moldova curriculumul la disciplina școlară Informatică păstrează module de programare, aparent rudimentare, pentru profilul umanist. Acest fapt permite absolvenților de liceu, profil umanist, continuarea studiilor la facultăți cu profil informatic sau ingineresc, atât în universitățile naționale, cât și peste hotare.

1.2. Informatica - domeniu competițional de performanță

Informatica a fost introdusă în programul de instruire preuniversitar în calitate de disciplină de studii, în anul 1985. Programul de studii la disciplină a fost de la bun început orientat către studierea conceptelor informatice fundamentale: structura sistemelor informatice și principiilor de bază a programării. Acest fapt a permis plasarea disciplinei Informatica în câmpul științelor reale. În consecință, disciplina a beneficiat de oportunitatea organizării concursurilor școlare de diferite nivele, la fel ca și celelalte științe.

Prima olimpiadă de nivel național a fost organizată în Republica Moldova în anul 1987 [62]. Pentru că în acea perioadă în instituțiile de învățământ preuniversitar practic nu existau

calculatoare, primele ediții ale Olimpiadei erau organizate în baza unor probe teoretice, la fel ca și concursurile matematice. Subiectele de concurs derivau din diverse compartimente ale matematicii aplicate, informaticii teoretice, logicii. De la ediție la ediție componenta algoritmică devine tot mai pronunțată, subiectele se orientează spre componenta practică – identificarea modelelor informatice de rezolvare a problemelor din diverse domenii.

Apariția calculatoarelor în școli stabilește începutul unei etape calitativ noi în organizarea concursurilor informatice. Devine posibilă testarea practică a implementării soluției teoretice într-un limbaj de programare și testarea acestei soluții. Practic imediat apare și primul sistem de evaluare automată a soluțiilor.

Începând cu anul 1996 elevii din Republica Moldova participă la Olimpiadele Internaționale de Informatică. Aceasta a impus în următorii ani ajustări ale modelului de organizare pentru Olimpiada da Republicană la Informatică. Este un punct de referință, pornind de la care se observă o dezvoltare continuă a tuturor componentelor structurale ale Olimpiadei Republicane: componente științifice (probe de concurs, evaluare), organizatorice (activități culturale, de comunicare, sesiuni științifice), tehnice (calculatoare, rețea).

În aceeași perioadă în școli se implementează modelul utilizării individuale a calculatoarelor la lecțiile de informatică (1 elev – 1 calculator). În rezultat crește numărul elevilor pasionați de informatică, în particular – de programare. Crește motivarea participanților la Olimpiadă, atât la nivel național, cât și local. Premianții Olimpiadei dispun de facilități la testările naționale, admitere la facultățile de profil din țară, beneficiază de diverse premii, concurează pentru participarea la concursurile internaționale de Informatică.

După anul 2000 are loc finalizarea procesului de modelare a principiilor organizatorice, științifice și de evaluare, acceptate ulterior prin regulamente și alte acte normative ale Ministerului Educației. Olimpiada la Informatică „se maturizează” și se dezvoltă în continuare în conformitate cu Curriculumul Național la disciplina Informatică și principiile olimpice, promovate de Comitetul Organizatoric IOI. Începând cu anul 2004 Moldova devine membru al comunității Țărilor Balcanice, organizatoare a BOI, iar în anul 2007 organizează pentru prima dată Olimpiada Balcanică la Informatică.

Anul 2017 are o importanță deosebită pentru informatica competitivă din Republica Moldova. În acest an țara găzduiește cea de a XXV-a ediție a Olimpiadei Balcanice la Informatică, la care participă douăsprezece echipe din țările Balcanice și țări invitate. În același an Reprezentanții Consiliului Olimpic la Informatică participă la inaugurarea unui nou concurs internațional de programare - Olimpiada Europeană la Informatică pentru Juniori - pentru care în anul 2024 devine și gazdă [63].

Olimpiada Națională la Informatică a generat în ultimii ani o serie de concursuri locale, organizate de instituțiile de învățământ: concursuri de programare individuale și pe echipe; concursuri on-line; concursuri de utilizare TIC; competiții în domeniul roboticii, etc. Astfel, numărul elevilor pasionați de diverse domenii ale Informaticii, crește continuu.

În mod special urmează să fie menționat concursul de programare, „The Game of Code”, organizat de Clubul de Informatică al liceului teoretic Orizont și Universitatea de Stat din Moldova. Ajuns la opt ediții realizate, concursul este suficient de matur pentru a fi considerat unul dintre factorii principali, care generează noi talente informatice în rândul elevilor de gimnaziu, contribuind în cea mai mare măsură la formarea comunității informatice din Republica Moldova. Tradițional, concursul este organizat în perioada sărbătorilor de iarnă, în regim on-line, cu un format și număr specific al problemelor propuse – circa 10 probleme de diferită complexitate, fiind perceput în mare parte ca un maraton de programare, chiar dacă timpul rezervat pentru rezolvarea problemelor este destul de scurt – patru ore astronomice.

Perioada pentru desfășurarea „The Game of Code” și formatul acestuia sunt alese în mod conștient – pentru a „injecta” elevilor, în mediu cu o lună înaintea concursurilor oficiale, mai multe probleme interesante pentru rezolvare, teme relevante pentru discuții, tehnici și soluții neordinare.

Olimpiada Republicană la Informatică este organizată anual, pe parcursul anului de studii: septembrie – mai. Desfășurarea Olimpiadei este organizată de Consiliul Olimpic la Informatică, componența căruia este aprobată anual prin ordinul Ministrului Educației și Cercetării. Concursul se desfășoară în trei etape: local, regional, național (Figura 1.1).

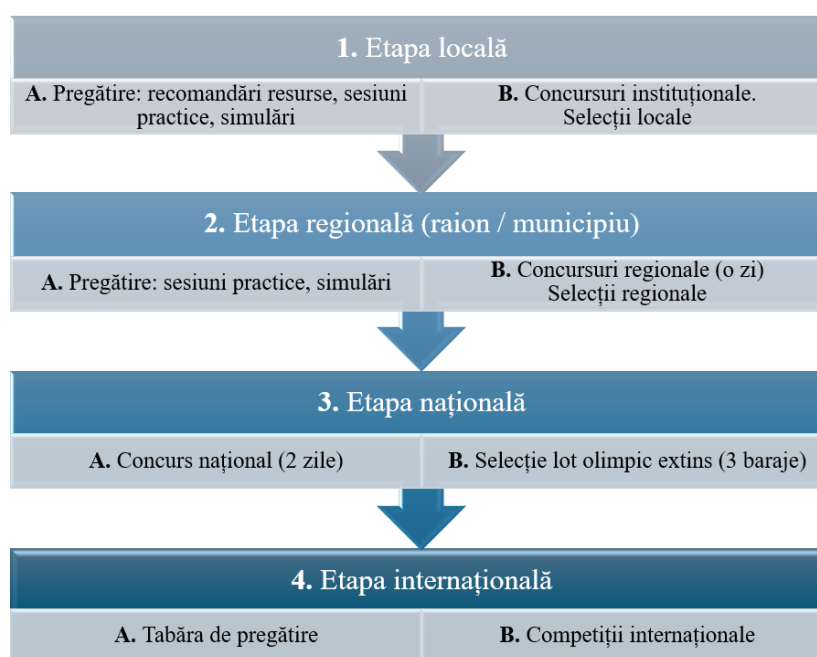


Figura 1.1. Etapele de desfășurare a olimpiadei naționale la Informatică și pregătire a lotului olimpic pentru concursurile internaționale

Categoriile de vârstă ale participanților: la etapele locale și regională concursul se desfășoară pe clase separate. La etapa națională sunt prezente patru categorii distincte: gimnaziu (clasele până la a 9-a inclusiv) și liceu – separat clasele a zecea, a unsprezecea, a douăsprezecea.

Evaluarea probelor în cadrul ORI este realizată cu ajutorul sistemelor informatice automate, ceea ce implică un model special de pregătire a problemelor de către autori. În calitate de sistem de evaluare este utilizată versiunea localizată a sistemului, utilizat pentru evaluare în timpul Olimpiadei Internaționale de Informatică – CMS. Fiecare problemă este evaluată cu baterii de teste de diferite nivele de complexitate și dimensiuni, care verifică atât soluția generală, cât și calitatea soluționării cazurilor extreme. Testele pot fi apreciate individual (fiecare test separat) sau pe gruppuri (subtask-uri).

Etapa locală a olimpiadei include perioada de pregătire a elevilor: septembrie – decembrie, dar și concursurile interne pentru selecția echipei instituției de învățământ – ianuarie. (Figura 1.1, etapele 1.A - 1.B).

Etapa regională este organizată separat în fiecare unitate administrativă. La etapa regională participă echipele instituțiilor de învățământ, selectate la etapa locală. În baza rezultatelor obținute la etapa regională se formează echipele pentru etapa națională. În municipii concursul poate să se desfășoare în două tururi, pentru a asigura o selecție mai riguroasă a echipei pentru etapa următoare. (Figura 1.1, etapele 2.A - 2.B).

Echipele pentru etapa națională se formează după principiul mixt: număr fix de participanți (4) + premianții Olimpiadei Naționale din anul precedent (variabil).

Numărul total de participanți la Olimpiada Națională poate varia de la an la an. Statisticile pentru perioada anilor 2017 – 2025 atestă o variații nesemnificative a numărului total de participanți pentru acest interval, care ar putea fi induse și de pauza de doi ani (2020, 2021) în organizarea competiției naționale (Figura 1.2).

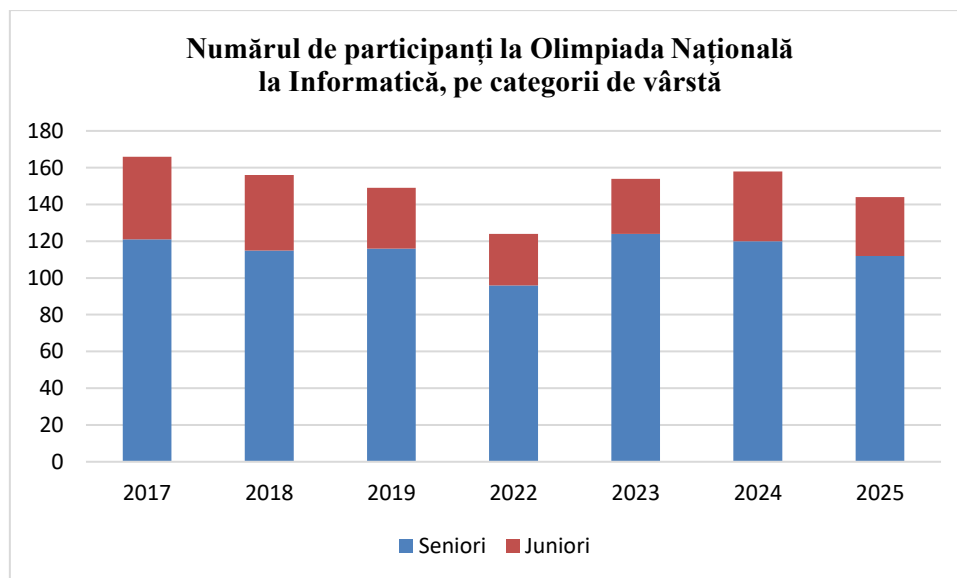


Figura 1.2 Numărul total de participanți la Olimpiada Națională la Informatică.

Micșorarea numărului de participanți pentru anul 2022 este motivată de lipsa de la Olimpiadă a premianților anului precedent, acesta fiind considerat 2019

Etapa națională. Participă echipele selectate la etapa regională, dar și premianții olimpiadei naționale din anul precedent. Etapa durează tradițional patru zile, două dintre acestea fiind rezervate pentru rezolvarea probelor de concurs. După finalizarea sesiunii de concurs participanții pot participa la o sesiune de autoevaluare, unde își verifică propriile soluții. În cazul apariției contestațiilor, acestea sunt prezentate Consiliului Științific al Olimpiadei. După epuizarea probelor de concurs și a contestațiilor are loc prezentarea soluțiilor oficiale. (Figura 1.1, etapa 3.A)

Premianții etapei naționale formează Lotul Olimpic Național la Informatică. Dintre ei este selectată una sau două Echipe Naționale la Informatică, care participă la competițiile internaționale. (Figura 1.1, etapa 3.B). Pregătirea către participările la competițiile internaționale oficiale începe cu o tabără de zece – douăsprezece zile, pe parcursul căreia participanții ascultă lecții teoretice, simulează concursuri virtuale, urmează un program cultural și de pregătire psihologică.

O importanță aparte o are tipologia subiectelor, propuse pentru rezolvare în timpul competiției naționale. S-a menționat deja că acestea sunt separate și diferențiate pe clase. Analiza metodelor de rezolvare a subiectelor propuse în perioada 2019 – 2025 la Olimpiada Națională², indică dominația problemelor de prelucrare elementară a numerelor și tablourilor (căutare, numărare, verificare proprietăți), urmată ca număr de apariții de problemele ce presupun o abordare dinamică, sau, în cazul restricțiilor mici – metoda reluării. Este foarte important, deja la primele experiențe naționale, participări să cunoască, să selecteze și să programeze algoritmi eficienți de sortare:

² Sursa datelor – arhiva digitală a Agenției Naționale pentru Curriculum și Evaluare (ance.gov.md)

problemele de sortare directă, împreună cu cele care folosesc metoda lacomă (Greedy), apar aproape tot atât de frecvent ca și problemele de prelucrare a numerelor și tablourilor (Figura 1.3).

În general, problemele sunt adaptate la nivelul de vârstă al participanților, metodele recomandate de rezolvare sunt ușor identificabile, clasele de probleme – corelate cu modulele opționale de programare din Curriculumul Național. Totuși, necesitatea abordărilor interdisciplinare este vizibilă deja la această etapă: sunt necesare cunoștințe elementare din teoria grafurilor, geometria computațională (matematica), înțelegerea esenței unor fenomene naturale (propagarea undei - fizica).

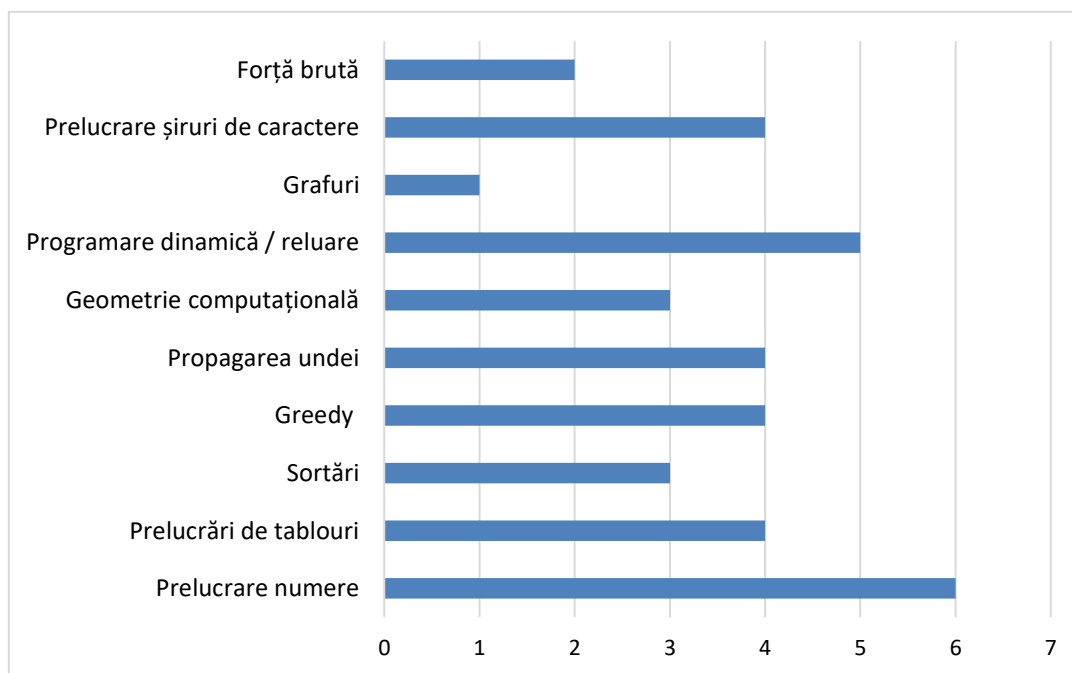


Figura 1.3. Tipologia problemelor propuse pentru rezolvare participanților la Olimpiada Națională, clasele de gimnaziu, în perioada 2019 – 2025

Repartizarea subiectelor în clasele a zecea – a douăsprezecea după metodele de abordare este diferită de cea prezentată anterior. Crește numărul de metode care pot fi folosite pentru rezolvare, domenii – sursă a situațiilor de problemă, complexitatea metodelor. Crește și rolul cunoștințelor și abilităților neinformatică, pe care la au participanții la concurs. Un exemplu elocvent în acest sens este problema „Hedge Fund” [64], propusă elevilor claselor a douăsprezecea la competiția din anul 2023. Problema are, indiscutabil, o soluție de natură informatică, dar vine din domeniul financiar, și este dificilă de rezolvat pentru elevii care nu au ascultat un curs de Economie sau nu au abilități puternic dezvoltate de creare a modelelor matematice. Din 41 de participanți cărora li s-a propus problema, doar șase au obținut punctaje parțiale sau integrale, ceilalți nici nu au încercat să o abordeze în timpul concursului.

În general, cele mai des întâlnite tipuri de soluții pentru problemele de concurs la Informatică în clasele zece - douăsprezece sunt cele de programare dinamică, algoritmi pe grafuri, metode de maximizare / minimizare, soluțiile euristice, apoi tehnicile standard de prelucrare a structurilor de date, inclusiv a arborilor binari (Figura 1.4).

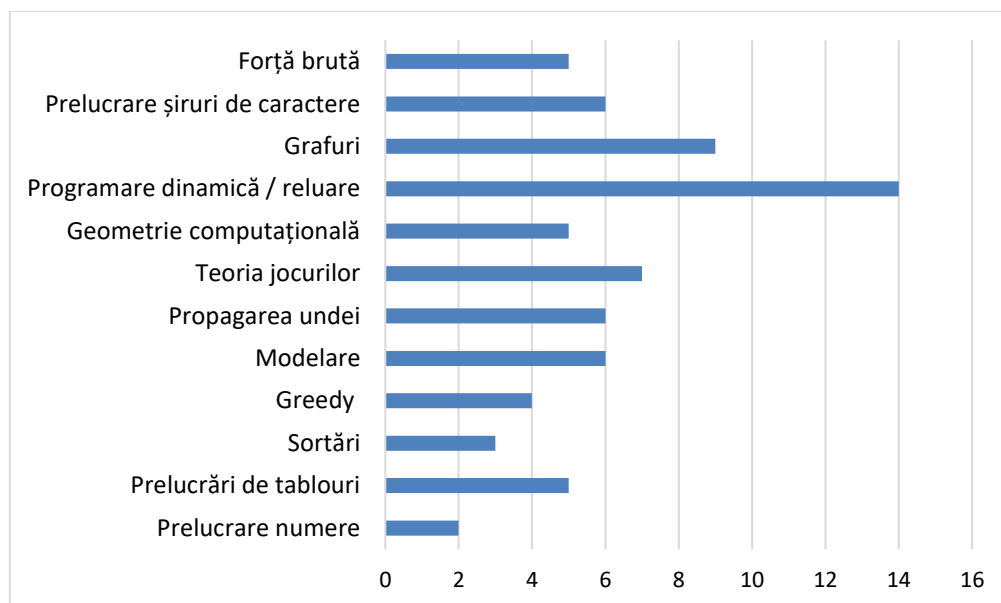


Figura 1.4. Tipologia problemelor propuse pentru rezolvare participanților la Olimpiada Națională, clasele X - XII, în perioada 2019 – 2025

Analiza rezultatelor obținute de elevi la Olimpiada națională la Informatică în perioada anilor 2017 – 2025 permite și observarea unor legități specifice, care au stat la baza prezentei cercetări:

1. Existența unor zone „neinformatică” care prezintă o lipsă permanentă a rezultatelor elevilor. De exemplu, raioanele Briceni, Edineț, Ocnita, Cimișlia, Ștefan-Vodă, Vulcănești, Basarabeasca. Pentru aceste regiuni au fost verificate în mod special rezultatele la ORI până în anul 2014 inclusiv. Sursele oficiale nu pot pune la dispoziție rezultatele ORI din anii precedenți anului 2014.
2. Prezența unor centre de performanță geografice, cinci la număr: Chișinău, Bălți, Tiraspol, Nisporeni, Drochia, care se evidențiază prin numărul și frecvența obținerii premiilor la ORI. Centrul de performanță Chișinău este, de facto, un metacentru, format din cinci centre de performanță instituționale: Liceul teoretic „Orizont”, Centrul de Excelență în Informatică și Tehnologii Informaționale, Liceul teoretic „M. Eliade”, Liceul teoretic „N. Milescu-Spătaru”, Liceul teoretic republican „Aristotel”;

3. Independența rezultatelor obținute de fiecare din centre în raport cu elevi concreți: în fiecare dintre centrele enunțate se obțin rezultate la nivel național în fiecare an, chiar dacă generațiile de elevi care parcurg ciclul liceal sau gimnazial se schimbă;
4. Diferențierea centrelor apare la etapa participării la competițiile internaționale: chiar dacă echipa națională este formată din reprezentanți a câtorva centre de performanță, în majoritatea absolută a situațiilor de concurs internațional, rezultatele sunt obținute de elevii, care reprezintă unul și același centru de performanță instituțional.

O altă serie de observații, efectuate în baza subiectelor olimpiadelor internaționale și regionale, duce la următoarele totalizări:

1. Se folosesc frecvent problemele interactive – problemele în care interacțiunea între participant și sistemul de evaluare se exercită nu o singură dată pentru o problemă, ci de multiple ori, în funcție de soluție, date și cerințe;
2. Formulările problemelor devin mai specializate. În consecință, cunoștințele de bază din diverse domenii devin mai necesare, la fel – capacitatea de a opera sigur și eficient cu modele matematice complexe.
3. Metodele de rezolvare sunt „ascunse” în fabula problemei. Capacitatea de a înțelege expres un text literar devine nu mai puțin importantă decât abilitățile informatice sau matematice.
4. De cele mai multe ori soluția optimală se obține combinând câteva metode informatice clasice împreună. De exemplu, utilizarea unui algoritm eficient de sortare poate îmbunătăți timpul de lucru al metodei programării dinamice, utilizarea arborilor binari tip heap poate îmbunătăți timpul de lucru al algoritmului Dijkstra, pentru calcularea tuturor distanțelor minimale într-un graf.
5. Restricțiile problemelor sunt de așa natură, încât soluțiile integrale nu pot fi obținute altfel decât cu optimizări complexe a memoriei și numărului de operații, ceea ce implică utilizarea structurilor de date avansate, de exemplu: arbori Fenwick, arbori de intervale, heap-uri și alte structuri de date avansate.

1.3. Abordarea interdisciplinară în studiul Informaticii

În calitate de știință aplicativă Informatica a apărut din necesitatea de a automatiza calculele aferente proceselor de cercetare în științele exacte - matematică, fizică, chimie; științele naturii – biologie, geografie, geodezie, meteorologie; științele sociale – sociologie, antropologie, geografie socială; științele economice. Deci este corect de a studia elementele fundamentale ale Informaticii, pornind de la situațiile, fenomenele și problemele reale, preluate din domeniile enumerate anterior și adaptate la nivelul de cunoaștere al elevilor din învățământul preuniversitar. În majoritatea lor, științele, problematica cărora a dus la dezvoltarea Informaticii, se regăsesc în disciplinele școlare, atât la treapta gimnazială, cât și la treapta liceală. Combinarea și integrarea în cadrul orelor de Informatică a metodelor, teoriilor și cunoștințelor obținute anterior la lecțiile de matematică, fizică, biologie, chimie și alte discipline permite, pe o parte, înțelegerea mai largă a esenței problemelor rezolvate informatic, pe de alta – a rolului Informaticii ca știință și disciplină școlară, utilității instrumentelor informatice în activitatea cotidiană, indiferent de domeniul general de studii sau profesie. În contextul definiției interdisciplinarității, date de Klein Julie Thompson în "Interdisciplinarity: History, Theory and Practice" ca „...*abordare academică și de cercetare care implică combinația și integrarea metodologiilor, teoriilor și cunoștințelor din mai multe discipline pentru a înțelege și rezolva probleme complexe...*” [65] se poate afirma că Informatica, în calitate de disciplină școlară, este un produs interdisciplinar generat de științele reale și cele ale naturii.

Abordarea interdisciplinară în studiul informaticii presupune integrarea și aplicarea cunoștințelor din diverse domenii, pentru a dezvolta înțelegerea și abilitatea de a rezolva probleme complexe. Informatica nu este o disciplină izolată; ea produce instrumente de cercetare pentru alte științe și, în același timp, se dezvoltă prin rezultatele cercetărilor acestor științe, obținute cu ajutorul instrumentelor informatice. De exemplu, matematica oferă bazele teoretice pentru algoritmi și structuri de date, fizica contribuie la înțelegerea și optimizarea hardware-ului, iar biologia și chimia inspiră noi modele și soluții în domenii precum bioinformatica și simularea moleculară; - dar toate aceste domenii se pot dezvolta doar în contextul utilizării instrumentelor de cercetare informatice.

În contextul acceptării conceptului STEM /STEAM (Știință, Tehnologie, Inginerie, Matematică / Știință, Tehnologie, Inginerie, Artă, Matematică) în aria curriculară Matematica și Științe, în care se regăsește și Informatica, aceasta din urmă devine un element central prin instrumentarul său digital, care poate fi modelat în funcție de context, atât pentru dezvoltarea competențelor integrate, de exemplu – utilizarea instrumentelor informatice pentru modelarea fenomenelor fizice sau ale mediului, utilizarea metodelor matematice pentru crearea structurilor

de date informatice, care să analizeze probleme de comunicare, sociologice, a grupurilor de influență, criptografice sau de orice altă natură [66].

Domeniile computaționale ale științelor exacte

Anterior s-a menționat că informatica a apărut din necesitatea de a automatiza calculele, primii algoritmi de calcul au fost algoritmi matematici, iar matematica modernă nu poate exista fără asemenea domenii ca combinatorica, calculul numeric, geometria și algebra computațională. Reciproc, fără funcțiile matematice clasice – funcția logaritmică, funcția exponențială, funcțiile trigonometrice, funcția putere și radical - implementate în instrumentarul standard al limbajelor de programare de nivel înalt nu ar putea exista nici interfețele grafice, atât de confortabile în utilizare, nici modelările 3D, nici asistenții inteligenți – nimic din ceea ce se numește revoluție digitală în secolul XXI.

Un studiu de caz, care ilustrează importanța instrumentelor matematice în studiul Informaticii este analiza complexității metodelor de sortare. Problema analizată este ordonarea șirurilor numerice. Instrumentele matematice pot fi folosite chiar la etapa inițială, pentru a genera șiruri mari de numere, formate din sute de mii sau chiar milioane de numere cu valori aleatorii. La următoarea etapă, de implementare a sortărilor, se observă eficiența diferită pentru metode de sortare distincte. Prin analiză estimativă a numărului de operații cu ajutorul calculelor matematice se estimează complexitatea fiecărui algoritm și se ajunge la cunoscutele valori ale complexității - $O(n^2)$ pentru algoritmi de sortare prin metoda bulelor, metoda sortării prin selecție și metoda sortării prin inserție. Prin aceeași analiză matematică a numărului de operații se ajunge la complexitatea stabilă de $O(n \times \log(n))$ pentru algoritmul MergeSort dar și la complexitatea instabilă, care variază de la $O(n \times \log(n))$ la $O(n^2)$ a algoritmului QuickSort. Rezolvarea informatică a problemei de sortare permite înțelegerea mai exactă a noțiunii de creștere a funcției prin comparația resurselor de timp necesare pentru rezolvare cu algoritmi de complexitate pătratică sau liniar-logaritmică.

Un exemplu mai simplu ar fi aplicarea instrumentelor informatice pentru desenarea graficului funcției pe un interval, calcularea soluțiilor unei ecuații sau sistem de ecuații, integrarea numerică sau transformările geometrice simple ale corpurilor 3D – deplasarea, rotația după axe, apropierea sau îndepărtarea. Realizarea fiecărei dintre aceste operații se reduce la recalcularea valorilor coordonatelor carteziane sau polare a unui set de puncte în spațiu, ceea ce din punct de vedere informatic nu e altceva decât o înmulțire a matricelor.

Interacțiunea cu Fizica se manifestă în diverse moduri. Despre importanța electromagnetismului pentru existența calculatoarelor s-a vorbit anterior. Dar elementele fizice apar și în timpul rezolvării unor probleme informatice. De exemplu, dacă în studiul de caz descris anterior, s-

ar pune problema măsurării timpului de lucru al unui algoritm distinct, ceea ce este de facto o problemă fizică, schema soluției informatice ar avea o structură similară următoarei:

```
T1 = Apel funcție de preluare a timpului curent, în secunde;  
Apel funcție de sortare sortX()  
T2 = Apel funcție de preluare a timpului curent, în secunde;  
Afișare T2 - T1, convertit în secunde;
```

De menționat că funcția de preluare a timpului curent și transformare a acestuia în unități de timp clasice, la fel ca funcția de sortare pentru oricare din algoritmi de sortare urmează să fie elaborate de elevi, utilizând instrumentele disponibile pentru căutarea informației.

Fizica computațională, un domeniu relativ nou și în continuă dezvoltare al fizicii, oferă instrumente importante pentru modelarea sistemelor fizice complexe cu scopul studierii comportamentului lor în diverse condiții. Fizica computațională combină puterea de calcul a sistemelor informatice cu principiile fizicii, permite astfel să fie explorat și prezis comportamentul universului într-un mod care nu ar fi posibil altfel.

În calitate de exemplu clasic al aplicării instrumentelor Fizicii computaționale poate servi studiul de caz, care vizează modelarea mișcării corpurilor sistemului Solar.

Înțelegerea și predicția mișcării planetelor Sistemului Solar este o problemă fizică clasică. Legile lui Kepler oferă descrierea mișcării planetelor în jurul Soarelui, dar nu iau în considerare interacțiunile gravitaționale dintre planete. Acestea pot genera abateri de la orbite eliptice perfecte. Instrumentele Fizicii computaționale permit să fie simulate aceste interacțiuni și mișcarea planetelor să fie prezisă cu o exactitate mai mare.

Problema constă în prezicerea pozițiilor planetelor în Sistemul Solar pe perioade lungi de timp, ținând cont de interacțiunile gravitaționale dintre ele.

Abordare: construirea la prima etapă a modelului fizic, apoi – a celui informatic.

Modelul fizic:

- Se consideră Soarele și planetele ca fiind puncte materiale cu masele și pozițiile inițiale cunoscute (bazate pe observații).
- Se aplică legea gravitației universale a lui Newton pentru a calcula forțele gravitaționale dintre fiecare pereche de corpuri (Soare, planete).
- Se ignoră alte efecte, cum ar fi relativitatea generală (care devin importante doar pentru orbite foarte apropiate de Soare sau pentru perioade lungi de timp).

Algoritm:

- Ecuațiile mișcării ($F = ma$) pentru fiecare planetă sunt ecuații diferențiale care nu pot fi rezolvate analitic, prin urmare se utilizează un algoritm de integrare numerică (de exemplu, Euler) pentru a aproxima soluțiile acestora.
- Algoritmul discretizează timpul (în pași de timp) și calculează pozițiile și vitezele planetelor la fiecare pas, pe baza forțelor gravitaționale calculate la pasul anterior.
- Algoritmul va oferi un rezultat cu atât mai exact, cu cât pasul de timp va fi mai mic, generând o predicție mai bună, dar va solicita mai multe resurse de calcul.

Simulare:

- Se scrie un program care implementează modelul fizic și algoritmul de integrare numerică. Este folosit un limbaj de programare, care dispune de instrumente integrate pentru crearea unui model grafic.
- Se rulează simularea pe o perioadă de timp specificată, cu salvarea periodică a coordonatelor obiectelor sistemului Solar.

Analiza rezultatelor:

- Se vizualizează orbitele și planetele simulate în programul elaborat.
- Se compară pozițiile planetelor simulate cu pozițiile observate.
- Se evaluează acuratețea simulării și se ajustează parametrii pentru obținerea unor rezultate mai bune.

În general Fizica computațională se ocupă de câteva domenii importante: erori și exactitate în calcule, integrale definite și derivate, ecuații algebrice și transcendente, transformări Fourier și interpretări fizice ale acestora, calcularea secțiunii de aur, metodele Runghe-Kutta, ecuații diferențiale de diverse tipuri, Metoda Monte-Carlo, generarea lanțurilor Markov [67]. Prin urmare există suficient de multe puncte de tangență între Matematică, Fizică și Informatică, care să permită rezolvarea problemelor fizice prin crearea modelelor matematice și calcularea soluțiilor prin metode informatice.

Chimia modernă, la fel ca Fizica și Matematica, a dezvoltat un aparat de calcul modern, bazat pe puterea de procesare a sistemelor informatice și modelarea matematică. Chimia computațională are un impact semnificativ asupra cercetării și aplicării practice a rezultatelor teoretice. Ceea ce permite utilizarea instrumentelor informatice în diverse domenii ale chimiei este varietatea modelelor numerice (matematice) statice și dinamice pentru simularea mecanismelor chimice și structurilor moleculare complexe. Modelarea Informatică ajută la înțelegerea interacțiunilor la nivel molecular, ceea ce este esențial în dezvoltarea de noi substanțe chimice și compuși farmaceutici.

Un exemplu semnificativ, de o complexitate care nu depășește limitele curriculumului școlare la informatică este analiza eficienței (mai exact a evoluției eficienței) a unui medicament, în scopul determinării optimale a dozei administrate.

Problemă: Este pus la dispoziție un set de date, care corespund unor serii de administrări ale medicamentului experimental pacienților. Se cere determinarea perioadei de timp după administrarea medicamentului, când acțiunea acestuia este maximală.

Rezolvare: La prima etapă, seturile de date sunt folosite pentru a construi, cu ajutorul polinoamelor Lagrange, o funcție polinomială, care aproximează suficient de bine funcția de eficiență a medicamentului [47]. Fiind cunoscută formula funcției care aproximează funcția căutată, cu ajutorul tehnicii de căutare binară se determină „vârful” funcției discretizate cu acel grad de exactitate, care este solicitat de condiția problemei.

În general, cercetările în domeniul chimiei moderne generează cantități mari de indicatori numerici, ceea ce pune în prim plan aspectele informatice de prelucrare și stocare a volumelor mari de date. Revenind la aspectele informatice, este nevoie ca elevii să cunoască tehnicile de prelucrare optimă a volumelor mari de date și metodele de optimizare a algoritmilor, selectarea structurilor de date adecvate [68].

Chimia computațională se bazează pe algoritmi și metode informatice pentru a prezice proprietățile și comportamentul moleculelor. Aceasta reduce numărul de experimente fizice necesare și accelerează procesul de descoperire și design de noi materiale. Un alt aspect al aplicării tehnicilor computaționale în cercetările chimice este automatizarea funcționării echipamentelor de laborator. Astfel este realizat un control mai precis al experimentelor și crește gradul de replicare a acestora, ceea ce este crucial pentru validitatea cercetărilor. Aceeași modelare matematică (mai exact - informatică) este folosită pentru a modela structura și comportamentul moleculelor. De aici – prin funcții de interpolare – metode informatice de predicție a proprietăților moleculare: instrumentarul informatic poate fi folosit pentru a prezice proprietăți fizico-chimice ale moleculelor, cum ar fi energia, structura electronică, stabilitatea și reactivitatea [69].

În modele simplificate aceste elemente de modelare pot fi transformate în probleme și exerciții, care permit, în context de rezolvare a problemelor chimice să utilizeze instrumente informatice.

Biologia computațională este un domeniu multidisciplinar care utilizează metode și tehnologii informatice pentru a înțelege, modela și analiza fenomene biologice complexe. Acest domeniu combină biologia, informatica, matematica și știința datelor pentru a aborda problemele din biologie la nivel molecular, celular și sistemic. Aspectele esențiale ale biologiei computaționale se referă la analiza secvențelor genetice - studiul și interpretarea secvențelor de ADN,

ARN și proteine, facilitare identificării genelor, a mutațiilor și a relațiilor evolutive între organisme [70].

Din punct de vedere informatic rezolvarea acestor probleme de reduce la utilizarea algoritmilor standard de prelucrare a șirurilor de caractere: de la elementarele funcții de căutare a secvențelor de elemente în șir până la identificarea celui mai lung subșir comun. Toate aceste operații nu depășesc complexitatea standard a problemelor curriculare la Informatică pentru clasa a XII-a, conform curriculumului național.

Un alt domeniu de utilizare a instrumentelor informatice în Biologie este cel al modelării proceselor biologice: Se dezvoltă modele matematice și computaționale pentru a simula procese precum metabolismul, semnalizarea celulară, replicarea genetică sau evoluția populațiilor, iar mai apoi, prin modificarea unor parametri se stabilesc intervalele de valori optime pentru dezvoltarea sau eliminarea populațiilor.

În esență, biologia computațională permite să înțeleagă și să prevadă comportamente și funcții biologice complexe, accelerând descoperiri științifice și contribuind la dezvoltarea biotehnologiei, medicinei personalizate și a altor domenii conexe.

Beneficiile unei abordări interdisciplinare a disciplinei Informatica, indiferent de contextul aplicării acesteia – în instruirea formală, neformală sau informală sunt multiple. Unu: aceasta îmbunătățește capacitatea elevilor de a rezolva probleme complexe, prin utilizarea unor metode și cunoștințe diverse. Doi: stimulează creativitatea și inovarea, deoarece elevii sunt încurajați să gândească dincolo de limitele unei singure discipline. Trei: contribuie la formarea unei perspective globale și integrate asupra cunoașterii, pregătind elevii pentru provocările viitorului, dezvoltând concomitent două tipuri de gândire, cruciale pentru performanță: gândirea critică și gândirea sistemică. Utilizând instrumentele de cunoaștere interdisciplinare, elevii întotdeauna vor putea găsi răspuns la întrebările: *ce este?*, *este adevărat sau nu?*, *cum funcționează?*, *de ce?*

1.4. Concluzii la Capitolul 1

Informatica, una dintre cele mai tinere științe, datorită aspectului său aplicativ, menține un ritm de dezvoltare dinamic deja de mai mult de jumătate de secol. Acest ritm impune implicarea unui număr mare de cercetători, având cunoștințe complexe din domeniul științelor exacte: informatică, dar și matematică, fizică, biologie, chimie. O creștere permanentă a numărului lor se poate menține doar prin politici educaționale corecte și consistente de promovare a performanței informatice la nivel național și internațional. Analiza efortului conjugat al actorilor sistemului educațional în ceea ce privește menținerea și consolidarea performanțelor la disciplina școlară Informatica permite să fie formulate următoarele concluzii:

1. La nivel național, există formațiuni teritoriale întregi, în care disciplina Informatică performanța informatică nu este promovată sub nici o formă.
2. La polul opus se află câteva centre de performanță, localizate în instituții din municipiile Chișinău, Bălți și Tiraspol; raioanele Drochia, Nisporeni, Soroca, unde curriculumul la disciplina Informatică este valorificat integral, ceea ce se demonstrează prin rezultate permanente la competițiile naționale.
3. Între acestea se regăsesc regiunile în care performanțele naționale apar sporadic, necontrolat, fie datorită automotivației înalte ale unor elevi, fie pe motiv de implicare în proiecte neformale de învățare a algoritmilor cu mentori la distanță, fie datorită apariției unor noi cadre didactice, cu ambiții și competențe în domeniu.
4. Curriculumul la clasă pentru disciplina Informatică poate fi folosit cu succes pentru pregătirea către competiții de nivelul local, regional sau național. Pentru a obține rezultate la competițiile internaționale, urmează alinierea la un syllabus internațional, precum și sesiuni de pregătire pe seturi de probleme atent selectate de mentor sau propuse de o platformă autonomă de pregătire.
5. Creșterea continuă a nivelului de performanță poate fi obținută prin antrenarea elevilor în concursuri de antrenament neoficiale, realizate online sau on site. Aceste concursuri au rolul de a forma motivația internă pentru informaticienii începători, de a-i ajuta să devină parte a comunității informatice, de a crea contacte și grupuri informatice neformale, capabile să se extindă atât cantitativ, cât și calitativ.
6. Cadrele didactice au nevoie de un model de instruire neformal, care să permită creșterea nivelului de performanță a elevilor, astfel încât aceștia să devină competitivi la nivel internațional, indiferent de regiunea geografică din care vin.

7. Informatica deține un instrumentar, care permite rezolvarea problemelor din multiple domenii, în special al științelor exacte, prin crearea modelelor matematice ale acestora și transformarea ulterioară a acestora în modele informatice pentru calcul automat.
8. Științele exacte au dezvoltat domenii întregi pentru integrarea în Informatică. Acestea sunt compartimentele de calcul numeric, geometrie și algebră computațională în Matematică, Fizica computațională, Biologia computațională, Chimia computațională – toate orientate către modelare matematică și calcule automate ulterioare.
9. Problemele de Fizică, Matematică, Chimie și Biologie organizate pentru rezolvare informatică, pot servi ca punct de start pentru dezvoltarea proiectelor integrate STEM sau STEAM.

2. MODELUL CRYPTEX PENTRU INSTRUIREA DE PERFORMANȚĂ LA INFORMATICĂ: CADRUL CONCEPTUAL ȘI METODOLOGIC

2.1. Conceptul de instruire de performanță a elevilor la Informatică

Noțiunea de *instruire de performanță la disciplina școlară Informatica* a apărut în contextul pregătirii echipelor olimpice naționale pentru participarea la Olimpiadele Internaționale la Informatică sau pentru alte concursuri de programare internaționale.

Olimpiada Internațională la Informatică, la fel ca și majoritatea altor concursuri de programare pentru elevi sau studenți, este un concurs la care participanților li se propun pentru rezolvare individuală probleme bine definite cu restricții specifice de timp, memorie, număr de accesări a serverului de evaluare, sau alte restricții specifice. Conținutul problemelor propuse este de obicei de natură matematică sau logică, solicită cunoașterea unui set de algoritmi fundamentali, a unor structuri de date cu proprietăți specifice (neliniare), abilități de rezolvare a problemelor. Problemele sunt elaborate într-o manieră specifică, care presupune rezolvarea lor cu ajutorul calculatorului.

Limbajele de programare, permise pentru utilizare în cadrul concursurilor informatice sunt de obicei variațiile limbajului C: C, C++, C#. Unele concursuri permit utilizarea limbajelor clasice de programare pentru instruire din familia Pascal sau limbaje moderne din clasa Python.

Datele sunt preluate din sistemul de intrare standard / consolă, iar rezultatele - livrate în sistemul de ieșire standard / consolă, dacă numai condițiile problemei nu solicită alte modele de comunicare.

Evaluarea soluțiilor problemelor propuse de participanții la concurs este realizată de un sistem automat de evaluare, care, în baza Regulamentului competiției, poate considera în calitate de răspuns (soluții) fie ultima variantă trimisă de participant, fie cea care este apreciată cu cel mai mare punctaj.

Capacitatea de a rezolva problemele de concurs implică cunoașterea și abilitatea de a transpune în practică o serie de cunoștințe și competențe specifice, care de multe ori pot depăși prevederile curriculumului la disciplina Informatica [71]. Acestea se referă atât la capacitățile intelectuale, cum ar fi gândirea matematică, capacitatea de a dezvolta algoritmi, de a identifica și corecta sub presiune erori, cât și la abilitățile motrice, cum ar fi deprinderea de a tasta la viteză înaltă, fără a urmări vizual textul scris pe ecran.

Deoarece succesul informatic este un produs integrat, care depinde nu doar de abilitățile de scriere corectă a codului în baza unui algoritm descris în pseudocod, or aceasta este activitatea cotidiană a programatorilor în sectorul ingineriei software, dar și de cunoștințele competitorului în

domeniile conexe (în capitolul precedent s-a stabilit deja legătura strânsă cu fizica și matematica la nivel de baze științifice a Informaticii, cu celelalte științe exacte – la nivel de interacțiune interdisciplinară) procesul de instruire poate fi extrem de dificil, în special în lipsa unui program clar de avansare cognitivă, care să echilibreze achizițiile de cunoștințe informatice cu contextul matematicii, fizicii, chimiei și biologiei, în special a compartimentelor computaționale ale acestor științe.

Este complicată atât misiunea elevului, care urmează să-și construiască traseul de progres în spațiul informaticii competitive, cât și misiunea mentorului, care urmează să-i indice posibilele puncte critice în traiectoria de progres, trasee intermediare optime, ba chiar și să-l încurajeze, prin propriul exemplu. În consecință – efortul mentorului este nu mai mic, iar pe alocuri mai consistent chiar decât efortul celor instruiți de el, cerințele înaintate de activitatea de mentor sunt mult superioare față de nivelul de cunoaștere atât a disciplinei predate, cât și a subiectelor aferente și depășesc mult cerințele înaintate de sistemul educațional față de cadrele didactice.

Prin urmare, pentru a obține rezultate palpabile în domeniul Informaticii competitive, este necesar un sistem integru: mentor erudit, elevi motivați, mediu educațional favorabil, resurse de învățare și traiectorii de învățare atent și corect selectate. Toate aceste condiții nu pot să apară împreună, în urma unor împrejurări întâmplătoare, dar pot fi create în urma unor activități manageriale strategic organizate și realizate pe parcursul unor perioade de timp extinse – de până la 5 ani.

Instruirea de performanță la Informatică este așa dar, un proces complex, iterativ care presupune implicarea unui set întreg de actori ai sistemului educațional, începând cu structurile centrale de conducere (Ministerul Educației și Cercetării, Agenția Națională pentru Curriculum și Evaluare - acestea au rolul de comunicator internațional dar și organizator al participării echipelor naționale la competițiile internaționale), administrația instituțiilor de învățământ (cum a fost menționat anterior, misiunea administrației instituției este crearea ecosistemului în care se dezvoltă comunitatea informatică), profesorul la disciplina Informatica, care apare într-un rol mai puțin obișnuit – de mentor, îndrumător, dar care trebuie să depună un efort permanent pentru a păstra acest rol, și elevul care trebuie să antreneze maximal capacitățile sale de a analiza, experimenta și optimiza continuu pentru a atinge performanțe majore. Ea nu se reduce la capacitatea de a scrie corect un cod de program în baza curriculumului la disciplină, oricât de bun nu ar fi acesta.

Procesul de instruire a elevilor pentru performanțe în informatica competitivă este unul de durată, multicomponent și complex. Elementul de bază, pe care se „montează” componentele succesului este un syllabus de performanță la disciplină, modular, corelat atât cu curriculumul național

la disciplina Informatică, cât și cu syllabus-ul internațional de performanță în Informatică, dezvoltat de Consiliul Olimpic Internațional.

Necesitatea unui asemenea document este argumentată practic de toți promotorii naționali ai mișcării olimpice în domeniul Informaticii: [72, p. 50] [73, p. 193] [74, p. 122] [75, p. 125] [76, p. 35] [77, p. 136] [78, p. 146] [79, p. 5] [80, p. 135] [81, p. 170]. Primele încercări de elaborare a Syllabus-ului de Performanță la Informatică prin prisma Curriculumului la disciplina de studii în Republica Moldova au fost efectuate de autor în perioada 2011 - 2014 [60, pp. 74 - 79] [61] [52, pp. 53-56]. La acel moment Republica Moldova avea la activ 6 participări consecutive la Olimpiada Internațională de Informatică fără un rezultat relevant [82, p. 1]. Elaborarea acestor documente, însoțită de instruiri ghidate de acestea, a avut un efect imediat – la ediția Olimpiadei Internaționale de Informatică din 2014, Taipei, Taiwan, echipa Republicii Moldova a obținut prima medalie de bronz după pauza de 6 ani. Fluxul de rezultate (4 medalii de argint, 13 medalii de bronz, 3 mențiuni de onoare) a continuat la toate edițiile ulterioare, până la ediția 2024 inclusiv [82, p. p.1]. Pe parcursul anilor 2014 – 2024 Syllabus-ul de performanță a fost actualizat în baza analizei continue a subiectelor Olimpiadelor Regionale și Internaționale de Informatică, devenind un instrument de importanță majoră pentru pregătirea elevilor către olimpiadele și concursurile informatice, atât naționale cât și internaționale. Conținutul actual al syllabus-ului elaborat este prezentat în tabelul 2.1.

**Tabelul 2.1. Syllabus de performanță,
completare pentru curriculumul școlar, actualizat 2025**

Matematici aplicate (pentru pregătirea de performanță la Informatică)	
A. Grafuri – structuri de date abstracte	
Conținuturi	Activități recomandate:
• Noțiuni: <i>graf; vârf, grad, muchie, cale, ciclu.</i> • Reprezentări: <i>2D (matrice), 1D (liste)</i> • Clasificări: <i>grafuri conexe, planare, orientate, bipartite, inverse, complementare</i> • Operații: <i>adăugare / lichidare vârf, adăugare / lichidare muchie, BFS, DFS, intersecție, reuniune, diferență</i> • Algoritmi <i>Componente tare conexe (Kosaraju), arbori parțiali de cost minim (Kruskal, Prim), Drum minim (Dijkstra, Floyd);</i> • Construcții <i>mediana absolută și relativă, centrul absolut și relativ.</i> • Flux maxim: <i>algoritmul și teorema Ford Fulkerson.</i>	Rezolvarea problemelor tip, care necesită implementarea algoritmilor în studiu: – problema liderului; – problema podurilor din Königsberg; – problema comisului voiajor; – problema poștașului chinez; – problema amplasării optime a centrului de deservire (depozitelor); – problema minimizării lungimii rețelei; – problema distanței minime între două puncte pe graf; – problema repartizării optime a resurselor etc. Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online – în runde

• Ciclomatică: ciclul <i>Hamilton</i> și problema comisului voiajor , ciclul <i>Euler</i> și problema poștaşului chinez • Grafuri generalizate: multigrafuri, grafuri cu bucle pe noduri, grafuri bipartite • Arbori: frunze, diametru, păduri • Combinatorica grafurilor: relații numerice între elementele grafului	de concurs (sincron) și după (asincron, fără modificarea rating-ului)
--	---

B. Structuri de date arborescente

Conținuturi	Activități recomandate:
Arbori: • arbori cu rădăcină: rădăcină, părinte, fiu, strămoș, subarbore, arbore binar • arbori de căutare binari și proprietățile lor; • arbori roșu – negru; • heap –uri; • arbori de intervale (arbori de segmente); • arbori Fenwick (arbori indexați binar); • hash; • arbore de sufixări • operații pe arbori;	Implementarea operațiilor admise: pentru structurile în studiu – inițializarea structurilor; adăugarea și lichidarea elementelor; căutarea; parcurgerea. Rezolvarea problemelor tip care necesită utilizarea arborilor: – determinarea înălțimii unui arbore binar; – reprezentarea mulțimilor disjuncte; – construirea arborilor de compresie Huffman; – construirea arborilor asociați expresiilor aritmetice etc

C. Strategii algoritmice

Conținuturi	Activități recomandate:
Tehnici avansate de programare: – metoda „Branch and bound” – metoda „Divide et Impera”, utilizare avansată – metoda căutării binare – programarea dinamică (descriere generală, spre sfârșit, spre început, probleme-tip)	Rezolvarea problemelor tip care necesită aplicarea tehnicilor în studiu: – Turnurile din Hanoi; – înmulțirea eficientă (numere, matrice); – problema rucsacului; – transformări rapide Fourier; – arborii Quad; – acumularea punctajului maxim; – traiectorii distincte de deplasare (linie, plan, spațiu); – distribuirea cadourilor. Rezolvarea și evaluarea problemelor pe platformele de instruire de performanță online

D. Algoritmi fundamentali

Conținuturi	Activități recomandate:
Operații aritmetice cu numere lungi: (+ - * /) Algoritmi combinatorici: permutări și combinări; poziția elementelor; determinarea indicelui elementului; elemente combinatorice „vecine”; numere remarcabile. Algoritmi cu mulțimi: submulțimi; submulțimii cu indicele dat; indicele submulțimii; submulțimi „vecine”; partiții. Algoritmi de căutare: liniară, binară. Algoritmi eficienți de sortare: sortare rapidă, heap sort, merge sort, radix sort, linear sort.	Rezolvarea problemelor tip care necesită aplicarea algoritmilor în studiu: – adunarea și scăderea numerelor lungi; – înmulțirea eficientă a numerelor lungi; – generarea elementelor combinatorice primare (aranjamente, amplasări, permutări, combinări); – generarea elementelor combinatorice avansate (numere Catalan, numere Stirling, numere Bell); – generarea submulțimilor unei mulțimi, submulțimilor cu proprietăți date; – problema „ghicirii” numărului;

	– gestionarea structurilor de tipul heap (inserare, eliminare); – compararea metodelor eficiente de sortare etc.
--	---

E. Analiza avansată a algoritmilor

Conținuturi	Activități recomandate:
Algoritmi numerici: <i>descompunerea în factori primi, ciurul lui Eratostene, algoritmul Euclid, metoda Gauss pentru rezolvarea sistemelor de ecuații, algoritmi eficienți pentru ridicarea la putere, algoritmi eficienți pentru verificarea numerelor prime, logaritmare eficientă, extragerea eficientă a rădăcinilor, descompunerea eficientă în factori primi.</i> Algoritmi pe șiruri de caractere: <i>căutarea subșirului în șir: metode brute, algoritmul Knut-Morris-Pratt, șiruri periodice și ciclice, ajustarea optimă a șirului, algoritmul Bayer-Moore, generarea arborelui de sufixe, descompunerea în șiruri simple (prime), algoritmul Aho-Corasick.</i> Algoritmi pe grafuri: <i>mulțimea maxim independentă, cuplajul maxim (algoritmul de forță brută, algoritmul maghiar), colorarea grafului, P-centre, P-mediane, reductibilitatea problemelor.</i>	Rezolvarea problemelor: – calcularea factorialului pentru $n > 10$; – calcularea numărului de zerouri finale în factorial; – calcularea factorialului în baza k; – calcularea numărului de zerouri finale în factorialul în baza k; – determinarea cifrei de pe poziția k în $n!$; – problema de maximizare (minimizare) a funcției într-un set de restricții date; – șiruri palindromice și transformarea lor; – editarea șirurilor; – subșirul maxim comun a două șiruri; – subșirul maxim comun a n șiruri; – transformări a șirurilor periodice; – identificarea elementelor în șiruri periodice; – transformări a structurilor ciclice; – identificarea elementelor în structurile ciclice; – descompunerea șirurilor; – identificarea elementelor în șiruri cu proprietăți date; – acoperirea cu un număr minim (maxim) de elemente; – realizare paralelă a proiectelor; – amplasare optimă a n centre de deservire etc.

F. Algoritmi geometrice

Conținuturi	Activități recomandate:
Noțiuni fundamentale: <i>punct, segment, dreaptă, plan, sistem de coordonate.</i> Transformări de coordonate: <i>deplasări, rotații, transformări mixte, coordonate polare.</i> Intersecții: <i>coordonate omogene, intersecția a două drepte, intersecția a două segmente, intersecția segmentului și a drepte.</i> Înfășurătoarea convexă: <i>algoritmii de forță brută, algoritmul Graham, algoritmul Andrew.</i> Aria poligonului simplu: <i>triangularizarea, metoda trapezelor.</i> Algoritmi fundamentali: <i>cea mai apropiată pereche de puncte, cea mai depărtată pereche de puncte, triangularizarea poligonului simplu, determinarea nucleului poligonului, domenii cu proprietăți prestabilite.</i>	Rezolvarea problemelor tip, care necesită implementarea algoritmilor în studiu: – problema cadrării; – problema calculării ariei unui domeniu; – problema fugarului; – problema poligonului și diagramei Voronoi; – problema controlului teritoriului; – problema divizării figurilor plane în triunghiuri; – problema triangularizării Delone; – problema domeniului convex de arie minimă care va conține o mulțime dată etc.

Matematici clasice (pentru pregătirea de performanță la Informatică)

G. Aritmetica și Geometria

Aritmetica: <i>proprietăți ale numerelor întregi, operații cu numere întregi, aritmetica modulară (adunare, scădere, înmulțire modulo), numere prime, fracții, procente.</i> Geometria: <i>linii, segmente, unghiuri, figuri, poligoane și proprietățile lor (vârfuri, laturi, simple, convexe, domenii, arii), distanțe euclidiene, Teorema lui Pythagoras, Geometria în 3D, calcule de precizie cu numere reale, numere complexe, structuri conice, funcții trigonometrice</i>	Activitățile practice se includ în rezolvarea problemelor grupurilor A - F. Problemele și exercițiile matematice (pentru rezolvare analitică) pot fi propuse din mai multe surse tradiționale, de exemplu: [83], [84], [85], [37], [86], [87], [88]
--	--

H. Structuri discrete

Funcții: <i>surjective, injective, inverse, compoziția; relații de reflexivitate, tranzitivitate, echivalență, ordine lexicografică; seturi, produse carteziane, putere.</i> Logica: <i>operații, forme normale.</i> Tehnici de demonstrare: <i>noțiuni: implicație, conversie, inversie, contrapozitiv, negație, contradicție; metoda directă de demonstrare, contraexemplu, contrapozitie, contradicție, inducție, inducție puternică, definiții matematice recursive.</i> Bazele calculului: <i>coeficienți binomiali, principiul includerii – excluderii, Identitatea Pascal, teorema binomială</i>	Activitățile practice se includ în rezolvarea problemelor grupurilor A - F. Problemele și exercițiile matematice (pentru rezolvare analitică) pot fi propuse din mai multe surse tradiționale, de exemplu: [30], [89], [90]
---	--

I. Criptografie

Problema de bază a criptografiei: <i>coduri simetrice, probabilistice, de bloc</i> Elemente de teorie a numerelor: <i>funcția Euler, algoritmi lui Euclid, teorema chineză a resturilor, teoreme numerice</i> Sisteme de criptare cu două chei: <i>semnături electronice Rabin, Fiat-Shamir, El-Gamal, logaritmare discretă, codarea fără chei, sistemele RSA</i> Jocuri criptografice: <i>lacătul temporal, protecția de falsuri, poker la telefon.</i>	Problemele și exercițiile criptografice (pentru rezolvare analitică) pot fi propuse din mai multe surse tradiționale, de exemplu: [91], [92]
--	---

Fizica (pentru pregătirea de performanță la Informatică)

J. Electromagnetism

Electricitate: <i>sarcina electrică, curentul, câmpul electric, potențiale, electromagnetism, circuite electrice, circuite integrate, electronica, unde electromagnetice</i> Magnetism: <i>surse, tipuri, electromagneți, câmpuri magnetice în materiale, forța magnetică, dipoli, originea mecanic-cuantică a magnetismului</i> Optică: <i>geometrică, reflecții și refracții, lentile, superpoziții și interferențe, difracții, dispersii, polarizare, laseri, efectul Kapitza-Dirac</i> Electromagnetica computațională: <i>descrierea generală a metodelor, alegerea, ecuațiile hiperbolice Maxwell, aproximarea bipolară</i>	Rolul componentelor fizice în pregătirea de performanță la Informatică este, în primul rând, unul conceptual, necesar pentru înțelegerea principiilor de funcționare a sistemelor informatice, memoriei și dispozitivelor de calcul. Activitățile practice vor fi orientate către asamblarea
---	---

<i>discretă, metoda momentelor și elementelor de frontieră, Domenii de frecvențe în diferențe finite, metoda elementelor finite, tehnica de integrare finită.</i>	sistemelor informatice instructive. Elementele matematice de calculare a parametrilor sistemelor fizice complexe intră în zona de calcul diferențial și integral. Exerciții și probleme - [93], [94]
---	---

Syllabus-ul de performanță este un document care răspunde doar la întrebarea „Ce trebuie să cunoască elevul” pentru a obține rezultate relevante în cadrul competițiilor. Răspunsurile la întrebările „Cum să învețe?” și „Cum să aplice” rămân, aparent, deschise.

2.2. Modelul interdisciplinar pentru instruirea de performanță la Informatică - CRYPTEX

În continuare va fi descris un model complex, care generează un răspuns integrat la toate cele trei întrebări Ce trebuie să cunoască elevul? Cum să învețe? Cum să aplice cunoștințele și abilitățile obținute? Modelul va integra dinamic conținuturile curriculumului general la Informatică cu mai multe conținuturi motivaționale, adăugând și conținuturile syllabus-ului descris anterior. Poate fi aplicat și parțial – fie doar pentru formarea abilităților de învățare, fără exersări în situații de concurs, fie pentru cizelarea abilităților formate în condiții de concurs real sau simulat, fie doar pentru a accede la un nivel prestabilit de elev sau profesor.

Modelul dezvoltat este o generalizare a modelelor de instruire concentrice, care pornesc de la Benjamin Bloom [95] și Jerome Bruner [96], în combinație cu conceptul centrelor de interes, formulat de Ovide Decroly și Gerard Boon [97].

Atât modelele clasice, cât și cele derivate din ele modelează progresul educațional în contextul educației formale, având în calitate de obiectiv final formarea unui set predefinit de abilități și competențe specifice unei discipline școlare sau, mai general, unui domeniu de formare. Modul de dezvoltare ulterioară a acestora este determinat de preferințele și aptitudinile personale ale fiecărui individ într-un context social specific.

Instruirea de performanță, indiferent de domeniu, se orientează către formarea unui set diferit de competențe, extins față de cel oferit de instruirea clasică, atât în sens de varietate de subiecte, cât și în sens de grad de complexitate a studiului fiecăruia dintre ele. În cazul Informaticii, sau, mai exact a domeniului ei științific, cunoscut sub numele de Computer Science, care apare în calitate de știință aplicată, și în calitate de disciplină de studii relativ recent [98] [62], ca un descendent al Ciberneticii [99] se identifică ci cel de al treilea vector – de corelare a procesului de formare a competențelor specifice domeniului Informatica cu procesele de formare a

competențelor specifice pentru științele exacte clasice: matematica și fizica, dar și cu alte domenii de cunoaștere, care în mod direct sau indirect influențează progresul formării informatice.

Un alt aspect important, care modelează procesul instruirii de performanță este creșterea continuă a discrepanței între syllabus-ul de performanță pentru disciplina Informatică și curriculumul național. Acesta a fost puternic influențat în ultimele decenii de principiile de „aerisire”, „simplificare”, „umanizare” etc, care au erodat puternic componenta științifică, fundamentală a conținuturilor studiate în favoarea componentelor cu tentă tehnologică, utilitară. [51] [100]. Pe de altă parte, Syllabus-ul internațional de performanță pentru disciplina Informatică [101] s-a îmbogățit cu o serie de compartimente, preponderent cu aspect interdisciplinar. Acest fapt a fost menționat de autor în articolul *Instruirea de performanță la Informatică în contextul curriculumului general* [60]. Astfel, devine inevitabilă necesitatea de acoperire a golurilor de cunoștințe, abilități și competențe cu module ce țin de alte discipline exacte și chiar de alte domenii de studii, pentru a menține performanța elevilor la un nivel comparabil cu standardele internaționale.

Informatica este o știință aplicativă. Atât cercetarea, cât și învățarea Informatică capătă sens doar în cazul utilizării sistemelor informatice. Eficiența utilizării acestor sisteme implică nu doar cunoștințe specifice dar și abilități psihomotorii speciale, pornind de la capacitatea de a introduce rapid textul, folosind tastatura; cunoașterea și înțelegerea construcțiilor metalingvistice, utilizate în limbajele de programare; a principiilor logice, matematice și fizice în baza cărora sunt construite și funcționează aceste sisteme.

Astfel, se ajunge la ideea necesității unui model complex trei dimensional, care permite creșterea optimală a performanței informatice în funcție de varietatea de conținuturi, activități educaționale și timp, urmărind progresul ca un rezultat integrat, care depinde de ajustarea corectă și, evident individualizată, a factorilor primari a instruirii.

Analiza retrospectivă a unităților de învățare, modulelor și compartimentelor studiate de elevi permite identificarea a trei nivele de performanță, realizarea consecutivă a cărora se demonstrează prin rezultatele obținute la competiții locale (nivel A), naționale (nivel B) și internaționale (nivel C). Această repartizare pe nivele o folosesc și mulți experți internaționali, cum de exemplu Biserka Yovceva și Peter Petrov, Bulgaria [102] [103], Vladimir Kiruhin, Federația Rusă [71], [104], Kairosh Makishev, Kazahstan [105], Emanuela Cerchez și Marinel Șerban, România [106], Krzysztof Diks și Jakub Lacki, Polonia [107], Steven Halim și Felix Halim, Singapore [108]. Fiecare dintre nivelele de performanță reprezintă la rândul său o structură complexă de activități de învățare – teoretice și practice, repartizate în timp și divizate în etape (la fel, câte trei etape pentru fiecare nivel). Fiecare etapă următoare are o complexitate teoretică și practică mai înaltă decât precedenta și, de cele mai multe ori, este o dezvoltare logică a acesteia.

De remarcat că atât specificul conținuturilor, cât și specificul celor instruiți impune studierea aproape concomitentă a etapelor în cadrul nivelului de complexitate format de etapele 1-3, 4 – 6, 7 - 9 . Creșterea intensității studiului teoretic generează solicitări de instrumentar informatic adecvat. Acesta, la rândul său, cere pentru utilizare abilități teoretice din zona științelor reale și algoritmicii.

Fiecare elev, care se angajează în procesul de instruire competitivă are o motivație inițială proprie, dar și un anumit volum de cunoștințe, de cele mai multe ori tocmai din zona interdisciplinară, care pot modifica viteza și ordinea parcurgerii conținuturilor și activităților practice pentru fiecare etapă de instruire. Dacă de poziționat unitățile de învățare pentru fiecare etapă pe suprafața unui disc separat, parcurgerea individuală a etapelor va fi însoțită de „ajustarea” reciprocă a discurilor vecine, pentru un rezultat optimal. Metaforic, formarea unui elev cu performanțe internaționale este similară deschiderii unui dispozitiv de tip CRYPTEX, [109] în care discurile reprezintă etapele de formare, potrivirea corectă a cărora permite să fie obținute și rezultatele așteptate.

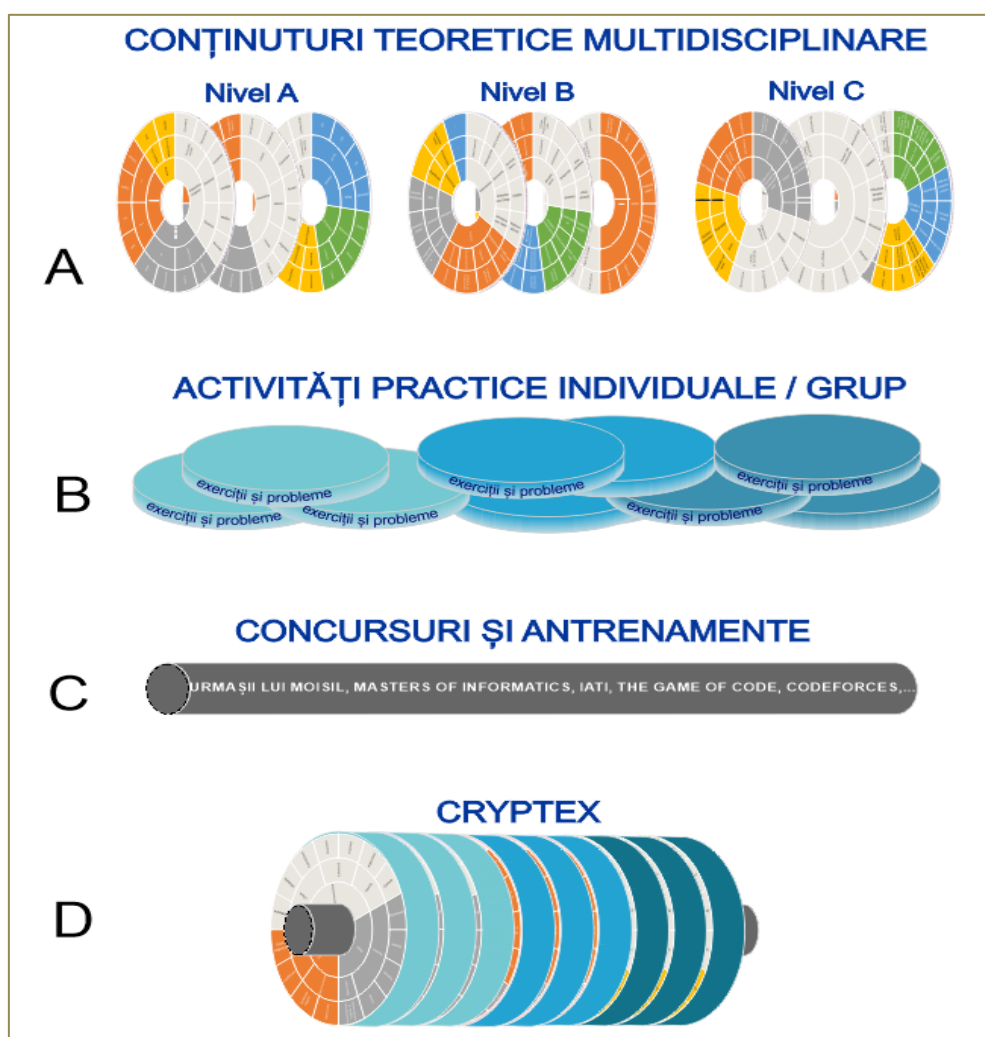


Figura 2.1. Schema generală a organizării procesului de instruire prin divizare în nivele și etape

Etapa de motivare inițială (A1)

Această etapă este una de pregătire a viitorilor informaticieni pentru înțelegerea corectă a specificului disciplinei, instrumentarului, comunicării și instruirii. Componentele de bază, trei la număr, identificate și testate în perioada 2000 – 2025 sunt:

Componenta matematică. Aceasta se axează pe studiul numerelor și a proprietăților lor, operațiilor, dar și a regulilor de formare a expresiilor aritmetice și logice. În procesul de studiu al numerelor sunt scoase în evidență proprietățile specifice ale acestora, în special pentru numerele întregi: numere pare și impare, pătrate și cuburi perfecte, numere prime, criterii de divizibilitate, șiruri de numere. Operațiile discutate – preponderent pentru prelucrarea numerelor întregi, operațiile relaționale și operațiile logice, cu aplicații în mediile de programare vizuale. În mod special – operațiile specifice împărțirii în numere întregi: împărțirea fără rest și operația modulo, sau restul întreg al împărțirii. La fel se introduc noțiunile de *adevăr* și *fals*, împreună cu operațiile logice specifice: ȘI, SAU, NU. Urmează expresiile aritmetice și logice, cu repetarea priorității îndeplinirii operațiilor, rolului parantezelor și modului de atribuire a rezultatului. Pentru exerciții și probleme cu numere reale se formulează și condiții de calcul sau reprezentare a rezultatului cu un anumit număr de cifre după virgulă. Se introduce noțiunea de reprezentare științifică a numerelor și se explică necesitatea acesteia. Tot aici se discută sistemele de numerație, regulile de trecere a numerelor dintr-un sistem de numerație în altul, rolul sistemului binar, reducerea operațiilor aritmetice în numere întregi la o singură operație. Pot fi introduse primele noțiuni și exemple de coduri numerice sau simbolice, exerciții practice de codificare sau reconstruire a unui text.

După studierea conținuturilor, ce țin de structurile fizice pentru modelarea memoriei și introducerea noțiunii de *bit* ca unitate de măsură a informației și *bit* ca unitate de memorie, se vor studia metodele de reprezentare a numerelor în binar pe un număr determinat de biți, inclusiv reprezentările complementare. La aceiași etapă se poate discuta despre adresele celulelor de memorie și metodele de calculare ale acestora, reprezentările hexazecimale, limitările impuse de reprezentările pe un anumit număr de biți.

Activitățile practice țin de dezvoltarea logicii în general și a logicii matematice în particular. Rezolvarea problemelor matematice și logice de dimensiune mică ($n \leq 4$) prin forță brută, identificând (și excluzând) situațiile imposibile, identificarea unor modele matematice simple după descrieri narative (în limbajul informatic - fabule) sunt parte a primei iterații de formare a gândirii algoritmice la elevii instruiți. Pe partea de sisteme de numerație se vor rezolva exerciții și probleme de transfer a numerelor dintr-un sistem de numerație în altul, transferurile directe între reprezentările binare, octale și hexazecimale. Povestirile, exercițiile și problemele matematice / logice pot

fi folosite atât în mod tradițional – în formă tipărită, de exemplu [110], [111], [88], [112], [87] sau ca și resurse digitale în spațiul digital: [113], [114], [115], [116].

Algoritmii și proprietățile lor se studiază pe parcursul întregii perioade alocate, accentul fiind pus pe descrieri, exemple și exerciții, care să permită generalizarea observațiilor și concluziilor, astfel încât formulările definițiilor și proprietăților să fie generate chiar de elevi. Metodele de reprezentare vor fi identificate de asemenea în urma rezolvării exercițiilor și a problemelor simple, cu soluții în iterații multiple. Resursele didactice de natură algoritmică pentru această etapă pot fi selectate din [115], [117], [118].

Componenta fizică. Se extinde asupra două domenii apropiate ale fizicii: electricitate și magnetism. Proprietățile magnetice ale materialelor permit înregistrarea, păstrarea în timp, și reproducerea informației. Elevii vor studia principiile de creare a unităților de stocare a informației și a dispozitivelor, care permit citirea / scrierea informației, principiile de codificare și transformare a informației în date și invers. După stabilirea legăturii între sistemul binar de numerație și stările magnetice ale celulelor de memorie se vor analiza metodele de codificare binară a informației pe dispozitive fizice, limitările impuse de aceste metode și modurile de evitare a acestor limitări. Un exemplu clasic în acest sens ar fi prezentarea sistemelor de codificare binară ASCII (pe 7 biți), ASCII extins (pe 8 biți) și UNICODE (pe 16 biți).



Figura 2.2. Componentele calculatorului Raspberry Pi: Cabluri pentru interconectare și conectare la monitor; tastatură; placă de sistem cu procesor și porturi montate, sistem de răcire, bloc de alimentare, memorii

În domeniul electricității elevii vor utiliza abilitățile de a modela circuite electrice, formate în cadrul lecțiilor la disciplina Fizică, stabilind legăturile între acestea și expresiile logice, din zona matematicii discrete. Modelând la prima etapă operațiile logice de bază prin circuite elementare, se trece la proiectarea ierarhică, cu modelarea circuitelor din ce în ce mai complexe, până la modelarea unității de calcul, a unității logice și a registrelor, care, împreună, formează dispozitivul aritmetic-logic (procesor) al oricărui sistem informatic. Realizarea teoretică a schemelor pentru componentele menționate ar putea duce la o dezamăgire și pierdere a interesului (parțială sau totală) din partea elevilor. Pentru a evita aceasta, se vor realiza activități practice de asamblare / dezasamblare a sistemului informatic. Pentru aceste activități se recomandă seturile Raspberry Pi. (Figura 2.2). Motivul este separarea în set a componentelor după funcționalități, (procesor, memorie RAM și ROM, porturi, etc) ceea ce simplifică înțelegerea structurii sistemului informatic și permite asamblarea unui prototip funcțional în timp redus, cu posibilitate de programare ulterioară a dispozitivului asamblat.

Componenta umanistică reprezintă o combinație de activități de învățare / motivare, vizuale sau narative, provenind din câteva domenii:

Resursele istorice au rolul de a demonstra dinamica dezvoltării sistemelor informatice, rolul acestora la diferite etape ale dezvoltării societății, posibilitățile de utilizare „netradiționale”, cum ar fi, de exemplu, construcția pianului mecanic, dispozitivelor de testare automată, consolelor de joc, etc. În calitate de resurse bibliografice pot fi folosite orice resurse veridice, în format tradițional (tipărit) sau digital. Pot fi recomandate: [119] – capitolele 9, 10, 11; [120] – integral, [46] – integral, [121] – integral, [122] – integral, [123] – capitole selectate.

Resursele literare. La etapa incipientă a instruirii înțelegerea importanței domeniului studiat are un rol extrem de important. Pentru început, incursiunea în istoria calculului automat, lecturile tematice și discuția ulterioară a operelor lecturate, studiile de caz privind tendințele actuale de dezvoltare a sistemelor informatice au în calitate de rezultat apariția unui interes constant și motivat pentru domeniul informaticii, ca știință, care oferă libertate și variate oportunități în societatea modernă. Lecturi recomandate: Stanislaw Lem, *Ciberiada*; Isaac Asimov, seria *Roboții*; Ray Bradbury, *Cronicile Marțiene, 451 după Fahrenheit*. În situații concrete lista de lectură se adaptează în funcție de preferințele elevilor instruiți, păstrând tematica generală în zona rolului sistemelor electronice pentru viitorul umanității, inteligenței artificiale, mediilor virtuale.

Nu trebuie ignorate nici resursele folclorului național sau internațional: situațiile bine cunoscute din copilărie capătă un alt sens în contextul informatic. Un exemplu clasic în acest sens este descrierea metodei de programare Greedy în „Fata babei și fata moșneagului”, sau fenomenului de furt al identității în „Scufița Roșie”.

Operele literare, indiferent de clasificarea lor, pot fi folosite în calitate de resurse pentru exerciții de identificare a unor algoritmi, descriși în limbaj uman – în cadrul activităților practice.



Figura 2.3. Repartizarea conținuturilor pe domenii la etapa de motivare inițială

Resursele lingvistice Pentru a forma vocabularul corect, utilizat de profesioniștii în domeniu, se recomandă utilizarea dicționarelor informatice specializate: explicative, de limbă engleză sau română și bilingve [124] [125] [126] [127]. Aceleași resurse pot fi folosite pentru a generaliza noțiunile de *alfabet* și *ordonare*. În consecință, primele exemple de sortare pot fi cele de sortare lexicografică, iar elementele primare ordonate – cuvintele și nu valorile numerice.

În paralel se formează primele concepte despre limbajele de programare și rolul lor în comunicarea cu calculatorul, sau, mai exact, în controlul sistemului informatic. În baza unor exemple simple pot fi demonstrate regulile de formare ale unor entități specifice limbajelor de programare, cum ar fi constantele numerice, identificatorii, expresiile de diverse tipuri. De aici – discuțiile pentru înțelegerea necesității limbajelor de programare: funcțiile lor specifice, structura,

elementele fundamentale, codul sursă și programul, procesele de transformare a codului în program și de îndeplinire a programului.

Totalizând cele expuse, conținuturile etapei A1 sunt descrise în figura 2.3 cu divizare pe disciplinele de studii.

Etapa instrumentală (A2)

La această etapă se formează deprinderile de utilizare corectă a instrumentarului informatic – limbajul de programare. Precondițiile accesului către această etapă sunt: cunoașterea noțiunii de program, a proceselor aferente creării programului, noțiunilor de compilare / interpretare, a tipurilor de date numerice și clasificărilor acestora, tipului de date simbolice (caracteriale), a noțiunii de memorie și a structurii ei.

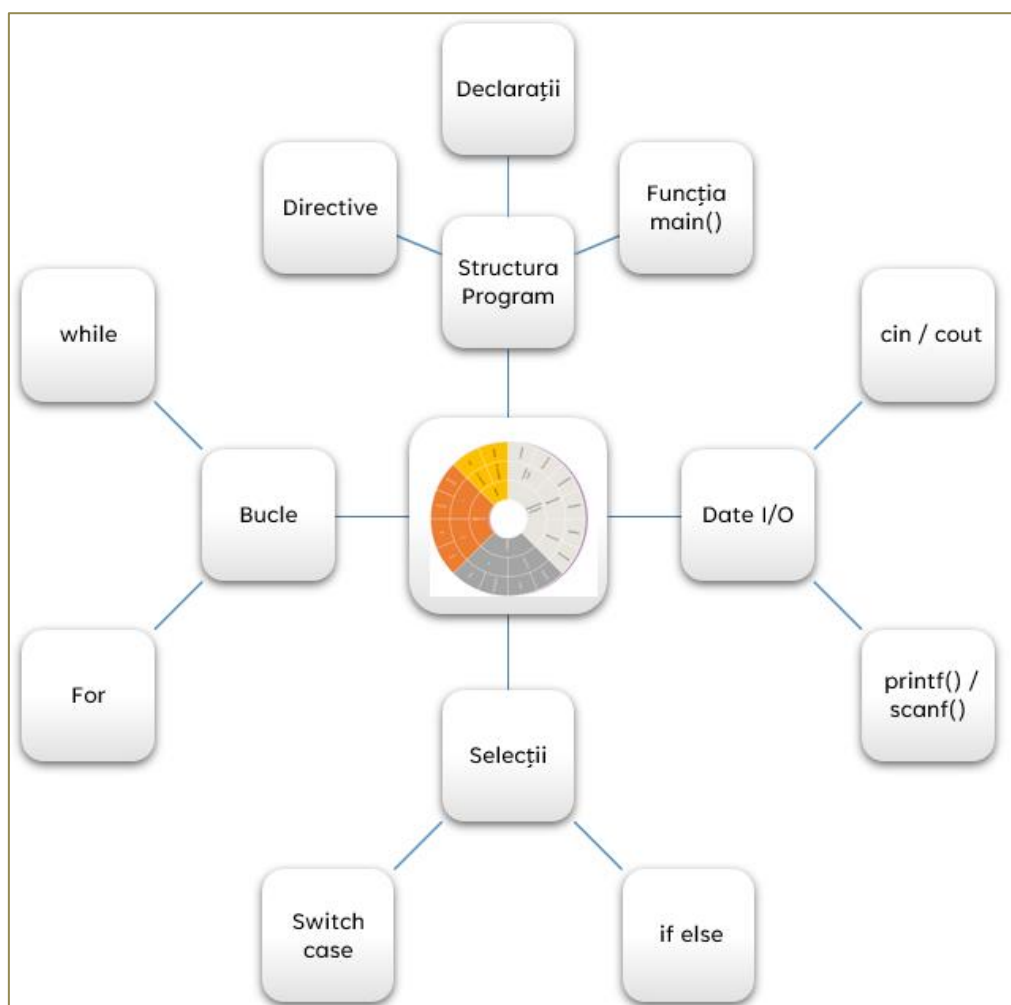


Figura 2.4. Conținuturile teoretice ale etapei instrumentale A2 - limbaje de programare

Formările încep de la studiul structurii unui program, a componentelor lui standard și a regulilor de organizare pentru fiecare dintre componente. Deoarece obiectivul final este performanța în cadrul competițiilor naționale și internaționale, limbajele de programare studiate se vor plia pe cerințele acestor competiții față de abilitățile de programare ale participanților. O analiză a

regulamentelor competițiilor de programare [128], [129] demonstrează că instrumentul universal pentru pregătire către competițiile de programare ale elevilor este limbajul C clasic sau C++. C++ este cea mai recomandată opțiune, din următoarele motive:

1. Dispune de un set extins de operatori (95 pentru versiunea 2020 în comparație cu 32 pentru C).
2. Are integrate librării pentru implementarea structurilor de date tip vector, tabele hash, arbori binari etc.
3. Are integrate librării pentru implementarea optimală a algoritmilor standard, cum ar fi sortarea rapidă, căutarea eficientă, generarea permutărilor, numărarea etc.

Prin urmare, urmează să fie studiată structura unui program C / C++, ceea ce implică asocierea practică cu noțiunile de directive, declarații, funcție principală. Prin activități practice se formează deprinderile de utilizare corectă a directivelor și spațiului de nume, declarare corectă a variabilelor, descriere a funcției principale. La această etapă este corect să fie utilizate nu doar manualele clasice pentru studierea limbajului C / C++, [130] dar și cele specializate, cum ar fi [106], [131], [132], sau resurse interactive verificate, în limba română [133].

Ordinea recomandată a studierii unităților de conținut coincide cu cea care este descrisă în curriculumul național pentru disciplina Informatica, este importantă intensitatea studierii acestora. Intervalul de timp de la primele programe scrise, până la utilizarea conștientă și corectă a tuturor construcțiilor de programare primare (instrucțiuni, selecții, bucle, subprograme) și a structurilor de date elementare (tablou liniar de numere, șir de caractere) din blocul A3 cu extensii în zona problemelor celebre B1, nu va depăși un semestru, preferabil – semestrul de toamnă, ceea ce va face posibilă participarea elevilor la competiții, începând cu primul an de instruire de performanță.

Traseele de precedentă pentru consecutivitatea studierii conținuturilor etapelor A2 – A3 – B1 sunt liniare:

Structura programului C / C++ (A2) → Citirea și afișarea datelor (A2) → Instrucțiuni (A2) → Formula lui Heron și alte formule de calculare a ariei, volumului, probleme fizice de transformare a unităților de măsură, de calculare a timpului, vitezei, distanței, temperaturii (B1) → Selecții (A2) → Problema selecției numărului maximal, minima, par / impar (B1) → Bucle (A2) → Algoritmul Euclid (B1) → Algoritmul Babilonean pentru calcularea rădăcinii pătrate (B1) → Tablouri liniare (A3) → probleme de identificare a elementelor tabloului cu valori date, maxime, minime (B1) → Funcții (A2) → probleme de identificare a elementelor tabloului cu valori, care posedă anumite proprietăți, acestea fiind determinate cu ajutorul unor funcții (B1) → Șiruri de caractere (A3) → probleme de identificare a simbolurilor și secvențelor de simboluri, numărări,

radieri, înlocuiri și înserări (B1) → probleme de selecție / calcul în bază de calendare (B1) → Probleme matematice cu șiruri de numere celebre (B1).

Resursele recomandate în calitate de surse de probleme și exerciții pentru activitățile practice care însoțesc această etapă sunt suficient de bine cunoscute. Numărul mic de ediții naționale este compensat de culegeri de probleme internaționale, special selectate pentru învățarea limbajului de programare în contextul pregătirii de olimpiadă. Un suport reușit în acest sens îl oferă *Pascal. Informatica. Culegere de probleme* de Andrei Braicov, un problemar, care poate fi folosit pentru activitățile practice în cadrul studierii oricărui limbaj de programare, nu doar a versiunilor limbajului Pascal [134]. Tot aici se regăsește primul volum al ediției *Programarea în limbajul C/C++ pentru liceu* de Emanuela Cerchez și Marinel Șerban [106], dar și *Informatica pentru gimnaziu. Culegere de probleme Pascal și C* de Doru Anastasiu [135].

Pentru descrierea algoritmilor, utilizați la rezolvarea problemelor la această etapă se va folosi fie limbajul uman, fie schemele logice.

Cunoașterea la această etapă a limbii engleze la un nivel suficient pentru citirea și înțelegerea unor probleme relativ simple va permite lansarea unei noi serii de activități practice – exercitarea în baza arhivelor de probleme de pe platformele specializate de pregătire / concursuri, cea mai recomandată fiind platforma Codeforces [136]. În lipsa cunoștințelor de limbă engleză la această etapă se vor folosi în același scop platformele de pregătire, a căror conținuturi sunt prezentate în limba Română, de exemplu – platforma InfoArena [137]. Se recomandă participările în afara rundelor de concurs, dacă acestea sunt anunțate, cu excepția rundelor pentru începători, cu selecție atentă a problemelor, fără depășirea complexității minime a problemelor pe platformă.

Structuri de date – în natură, societate și informatică (A3)

De cele mai multe ori rezolvarea informatică a problemelor presupune prelucrarea unor volume mari de date. Pentru una și aceeași problemă, în diferite condiții (perioadă de timp, zonă geografică, altitudine) variază doar valorile concrete ale datelor utilizate, nu și modul de organizare. Cantitatea mare de valori sau seturi de valori impune un mod de organizare a acestora, de exemplu temperaturile medii pe zile, pe parcursul unei luni calendaristice. Pentru a păstra relația între temperatură și dată, fiecărei valori măsurate i se pune în corespondență indicele zilei. Pe o hartă de teren cu zone unitare, fiecare zonă poate avea o mulțime de caracteristici: altitudine, tip de sol, vegetație, nivel de pericol seismic – valori diferite – structură identică.

În chimie structurarea corectă a datelor în formă de tabele hash permite identificarea operativă a compușilor. În Sociologie și alte științe umanistice tabelele pot stoca și organiza datele experimentale sau obținute în urma sondajelor – în format liniar sau tabelar. Mai mult, aceste

date pot fi organizate în seturi de natură diferită – un amestec de date numerice, simbolice, răspunsuri tip Adevăr / Fals – toate asociate unui singur individ discret.

Exemplele menționate vin să confirme definiția structurii de date ca un model de organizare și stocare a datelor, care asigură accesul rapid la date și realizarea eficientă a operațiilor asupra datelor particulare, fără a evidenția operațiile elementare efectuate [138] [29] [68].

Structurile de date sunt diferite. La fel, este diferit modul de alocare a memoriei pentru elementele structurii de date, modul de descriere a componentelor și acces la elementele structurii de date, varietatea și complexitatea operațiilor permise pentru fiecare tip.

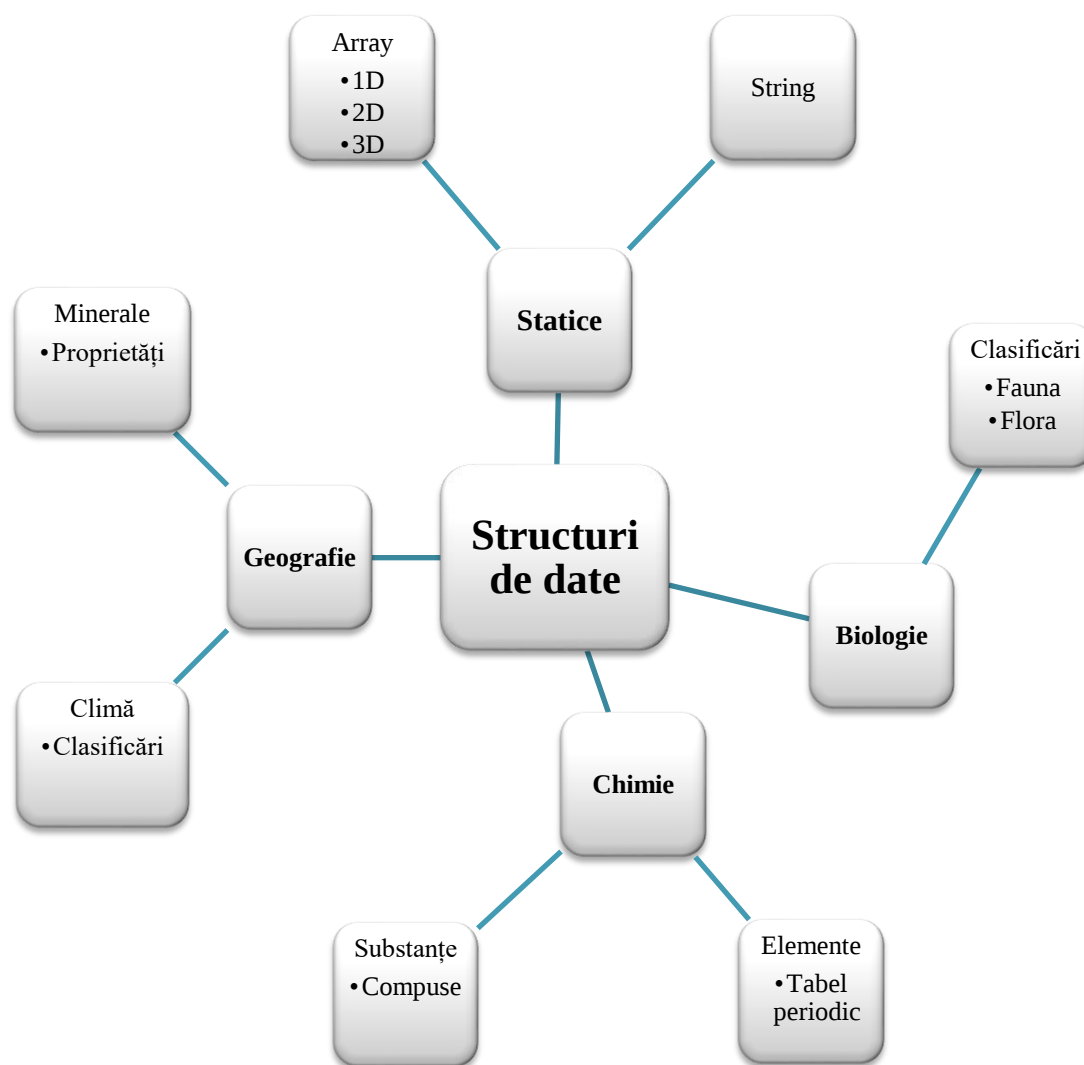


Figura 2.5. Structuri de date și clasificarea acestora după diverse criterii în funcție de domeniul de aplicare

Operațiile atribuite structurilor de date sunt diferite, atât ca mod de organizare, cât și în plan de complexitate. Dar, setul de bază al operațiilor studiat la etapa A3 este format din operații de inițializare a structurii de date, de introducere / inserare a elementelor sau valorilor acestora în structura de date, parcurgerea structurii de date, extragerea elementelor / valorilor din structura

de date. În structurile de date statice, în care spațiul de memorie este alocat o dată, la declararea structurii, nu pot fi implementate operațiile de excludere a elementelor ci doar a valorilor acestora, cu modificarea indexării. La fel – adăugarea elementelor nu este posibilă. În schimb pot fi adăugate valori în limitele spațiului de memorie, alocat inițial.

Studierea structurilor de date începe odată cu studiul bazelor programării (etapa A2) și continuă treptat, la etapele B1 – B3, C1 – C2.

Clasificarea structurilor de date este una standard [68], utilizarea practică a acestora începe cu cele mai simple: structurile statice, organizate liniar, cu indexarea elementelor. Această subcategorie ale structurilor de date poate fi introdusă imediat după sau în procesul de studiere a construcțiilor de programare fundamentale – buclelor cu contor, în baza relațiilor dintre valorile consecutive ale contorului buclei și indicii consecutivi a elementelor tabloului unidimensional și / sau a șirurilor de caractere.

Tablourile unidimensionale numerice. Structura și principiile de alocare a memoriei pentru acestea pot fi explicate pe etape, dar este important să se facă legătura între indexarea de la 0 a elementelor unui tablou liniar și modul de alocare a memoriei pentru tablou, în special – a formulelor de calculare a adresei fizice de memorie a fiecărui element al tabloului. Este un moment important, înțelegerea căruia presupune o abordare mixtă fizică – matematică pentru modelarea pașilor procesului de scriere / stocare / ștergere / a informației într-o structură de date informatică, și transformare a informației în date.

Problemele tip, cu care începe studierea tablourilor liniare sunt problemele de numărare, căutare a elementelor. Formularea lor poate fi atât abstractă, matematică, cât și cu ajutorul unui model ascuns într-o fabulă tematică: fizică, socială, etc.

Problemă-tip: din intrarea standard se citesc n valori numerice întregi, pozitive. Urmează să se determine valoarea maximală introdusă și numărul de apariții a acesteia în printre numerele în studiu (abstract). Sau aceeași problemă, reformulată: în procesul de observare a condițiilor meteorologice o stație meteo transmite zilnic către centrul de date valoarea medie a temperaturii măsurate. Urmează să se determine care a fost cea mai înaltă valoare a temperaturii zilnice medii, și de câte ori s-a observat această valoare pe parcursul perioadei de observație.

După studierea funcțiilor la etapa A2 se poate reveni la un nivel mai înalt de complexitate a condițiilor problemei, prin numărarea unor elemente cu proprietăți, care se identifică cu ajutorul unor funcții scrise de membrii grupului instruit.

După etapa problemelor de numărare pot fi propuse probleme de divizare a numărului în cifre separate și manipulare cu acestea: excluderea unei cifre pentru obținerea unui număr nou cât se poate de mare / de mic / care să aibă anumite proprietăți.

Probleme-tip: 1. *Un număr întreg, pozitiv se numește echilibrat dacă prima lui cifră coincide cu ultima. De exemplu, 1361 este echilibrat, 236 - nu. Scrieți un program, care determină, dacă un număr N este echilibrat sau nu. În cazul în care numărul este echilibrat, se va afișa cifra lui extremă, în caz contrar se va afișa -1* (Olimpiada zonală între colegii. Zona Centru-Sud. 2019).

2. *Cifra dominantă a numărului este cea mai mare ca valoare cifră din scrierea numărului. În numărul 12917 cifra dominantă este 9, în 23132 – cifra dominantă e 3. Scrieți un program, care determină cifra dominantă din număr.* (Olimpiada zonală între colegii. Zona Centru-Sud. 2019, sesiunea practică).

Un nivel mai înalt de complexitate se presupune a fi la problemele a căror condiții presupun modificarea poziției valorilor elementelor din tablou. Primele probleme, în acest context, urmează a fi cele de deplasare circulară a valorilor elementelor tabloului cu o poziție la stânga / dreapta, urmată de generalizare prin solicitarea de deplasare cu k poziții. Tradițional, cea de a doua problemă se rezolvă prin forță brută, repetând de k ori deplasarea cu o poziție. Or, tocmai aici este momentul de a lansa o investigație matematică și de a determina formula analitică pentru determinarea instant a poziției finale pentru fiecare valoare a tabloului, care inițial se afla în poziția $[i]$.

Înțelegerea mecanismului fizic și matematic de transfer al valorilor elementelor permite abordarea unor probleme și algoritmi fundamentali – algoritmi de ordonare (sortare) și algoritmi de căutare binară.

Se discută mai întâi noțiunea de ordonare, după care se identifică existența diverselor forme de ordonare, fiind evidențiată ordonarea cu chei numerice, ascendentă sau descendentă și ordonarea lexicografică, care apare în cazul elementelor simbolice. Deoarece ordonarea după chei numerice se implementează mai simplu la nivel de cod, studiul algoritmilor care rezolvă această problemă începe cu *Sortarea prin metoda bulelor*, sau - *BubbleSort*. Acesta este un algoritm intuitiv, ușor de demonstrat prin exemple din diverse domenii cotidiene de activitate. BubbleSort presupune parcurgerea consecutivă a perechilor de elemente vecine din secvența de numere și interschimbarea valorilor în perechile unde ordinea reală nu corespunde ordinii cerute. Numărul interschimbărilor efectuate la o parcurgere se contorizează. Parcurgerile se repetă până în momentul când numărul interschimbărilor la o parcurgere devine 0.

Explicarea în limbaj uman a algoritmilor devine din ce în ce mai complicată, odată cu creșterea numărului de operații distincte în algoritm – descrierea precedentă este un exemplu în acest sens. Se ajunge la momentul, când între etapa de citire și înțelegere a problemei informatice și etapa implementare în cod a soluției este necesară o descriere intermediară a algoritmului, pseudocodul devenind cea mai compactă metodă în acest sens [44].

Urmează studierea încă a două metode elementare de ordonare: metode selecției și metoda interclasării, descrise exhaustiv practic în toate manualele de studiu al algoritmilor.

Următoarele serii de exerciții propuse au scopul de demonstrație a diferențelor de rezolvare a unor probleme, aparent simple, pe tablouri mari, ordonate și neordonate. Pentru aceasta se va introduce inițial noțiunea de numere aleatorii, apoi se vor studia metodele instrumentale ale limbajului pentru obținerea unor secvențe de numere aleatorii. Se va reveni la problema căutării elementului cu valoare dată într-un tablou ordonat și se va introduce *metoda căutării binare*. În acest moment este necesară cunoașterea de către cei instruiți a noțiunii de *logaritm*, la necesitate aceasta va fi explicată de către mentor.

Din zona instrumentală a limbajului vor fi studiate funcțiile de fixare a timpului curent dar și metoda determinării timpului necesar pentru executarea unui fragment de cod. Aceasta va permite organizarea „competiției” în interiorul programului – măsurarea de către program a timpului necesar pentru identificarea prezenței unui element în tabloul neordonat și pentru aceeași misiune, în același tablou, ordonat prealabil.

O problemă semnificativă în studierea căutării binare este problema *Cea mai lungă secvență bună* [139], în care căutarea binară apare ca o metodă de realizare a unei etape a problemei de bază.

Șirurile de caractere. La studierea șirurilor de caractere sunt deja cunoscute noțiunile de tablou liniar, elemente ale tabloului, indici. Suplimentar, se introduc noțiunile de lungime a șirului de caractere și simboluri invizibile.

Primul set de probleme va urmări formarea abilităților de implementare a operațiilor – tip pe șiruri de caractere: determinarea lungimii șirului de caractere, compararea șirurilor, căutarea șirului Q în șirul S, radierea unui fragment de șir dintre doi indici, inserarea șirului Q în șirul S, începând cu un indice dat, concatenarea a două șiruri și divizarea șirului în două subșiruri după un anumit indice.

Probleme-tip: 1. *Nicu a primit un mesaj SMS. Mesajul conține litere ale alfabetului englez, cifre, spații și simboluri speciale. Determinați ultima literă (din dreapta) din mesaj. Mesajul se citește din input-ul standard. Lungimea mesajului nu depășește 1000 de caractere (Olimpiada zonală între colegii. Zona Centru-Sud. 2018).*

2. Compania StarSoft a lansat un nou server poștal, care oferă utilizatorilor accesul în baza unor parole variabile. Astfel, utilizatorul nu mai trebuie să țină minte o parolă exactă, ci doar o regulă, conform căreia să-și formeze o parolă validă ori de câte este necesar. Dorel are o regulă foarte simplă pentru formarea parolei: fiecare caracter următor din parolă (citit de la stânga la dreapta) trebuie să aibă un cod mai mare decât caracterul precedent. Parola nu poate conține spații. Suplimentar lungimea parolei nu trebuie să fie mai mică decât 8 și mai mare decât 64 caractere. Din neatenție Dorel uneori introduce și parole nevalide. Modulul de audit al sistemului păstrează lista de parole introduse. Scrieți un program, care va număra, câte parole din lista celor introduse de Dorel sunt nevalide (Olimpiada zonală între colegii. Zona Centru-Sud. 2019, sesiunea practică).

Următoarea etapă de instruire are scopul optimizării procesului de scriere a codului-program. Din zona instrumentală a limbajului – implementările tablourilor liniare prin tipul vector, automatizarea operațiilor de inserare a elementelor în poziții predefinite (început sau sfârșit), operații de radiere din vector, parcurgeri, căutări – tot setul pus la dispoziție de librăria specializată <vector>, dar și funcția de sortare automată sort().

Acești scop se va urmări și la introducerea tipului de date string și a librăriei de funcții pentru prelucrarea șirurilor de caractere <string>. Operațiile, realizate anterior prin funcții proprii devin accesibile în formă de funcții, componente a librăriei. Acestea pot fi accesate oricând este necesar din sistemul de asistență.

Tipul struct. Un tip de date, definit la necesitate în problemele în care se prelucrează obiecte de natură complexă. Punctul în plan poate fi interpretat ca o structură cu două componente – abscisa și ordonata lui; segmentul – ca o structură din două puncte, sau ca un set de patru numere, reprezentând nemijlocit coordonatele extremităților; o muchie a grafului – ca în set de doi indici ai vârfurilor pe care le unește, suplimentat de caracteristicile numerice proprii muchiei; date cu format specific descrise nemijlocit în librăriile limbajului de programare.

Utilizarea variabilelor simple de tip struct este ocazională și specifică. Un exemplu real de utilizare sunt variabilele tip time – time fiind o structură predefinită care conține în calitate de componente unitățile de timp, fiind descrisă în librăria <sys/time>. Descrierile exhaustive ale tipului struct a structurilor implementate, a operațiilor și a integrării tipului struct în tablouri sunt realizate în [130] [133] [106].

Printre primele probleme în care utilizarea tipului struct apare ca o metodă de optimizare a codului sunt problemele de sortare, în care obiectele cu multiple câmpuri urmează să fie ordonate după valorile unei singure sau câtorva chei (câmpuri). Utilizarea tipului struct permite interschimbarea concomitentă a tuturor valorilor câmpurilor obiectelor, și nu unul câte unul, cum,

de obicei, încearcă să procedeze adepții tablourilor liniare. Urmează în mod special să se atragă atenția despre posibilitatea utilizării funcției sort() și pentru tablourile liniare în calitate de elemente a cărora apar structuri. În acest caz este necesară descrierea unei funcții suplimentare, care stabilește cheia de ordonare (câmpul) și modul – ascendent sau descendent.

Probleme tipice pentru această etapă sunt și problemele de prelucrare a seturilor de puncte, figurilor și corpurilor geometrice, propuse de obicei pentru ilustrarea metodei de forță brută. În calitate de problemă rezolvată brut se poate propune determinarea celei mai depărtate sau celei mai apropiate perechi de puncte – cu condiția revenirii la soluțiile optime ale problemelor peste câteva etape.

Tablourile numerice 2D și tablourile de structuri 2D. Tablourile 2D sunt o generalizare a tablourilor liniare, în care poziția fiecărui element este determinată de doi indici – indicele liniei și coloanei la intersecția cărora se regăsește elementul – un neadevăr fizic, care este acceptat doar pentru că este foarte intuitivă înțelegerea tablourilor, reprezentate în formă de matrice cu un număr fixat de linii și coloane. În realitate un tablou 2D cu N linii și M coloane este un tablou liniar cu $N \times M$ elemente, în care secvența de celule de memorie este formal împărțită în N subsecvențe de lungime M. Din nou – numerotarea de la 0 a liniilor și coloanelor tabloului 2D este motivată de formula de calcul a adresei fiecărui element al tabloului, pornind de la adresa de bază x:

Adresa elementului $a[i][j]$ este $x + (i \times M + j) \times V$, unde V este volumul de memorie alocat unui element al tabloului.

Tablourile 2D și „extensiile 3D” sunt extrem de des utilizate, fiind un mediu de modelare perfect pentru situații în care se analizează hărți geografice, imagini și obiecte care se conțin în ele, spații pentru depozitare, labirinturi, table de joc,

Probleme-tip: 1. „Harta” *Sistemul automatizat de informare a Serviciului Situații Excepționale stochează datele despre structura regiunilor geografice în tablouri bidimensionale. Fiecare element al tabloului se referă la o regiune dreptunghiulară de teren, valoarea elementului indicând obiectul geografic dominant al regiunii: 1 - pădure, 2 – câmpie, 3 – râu, 4 – localitate.*

1	1	1	3	3	3	3
1	4	2	2	1	1	3
2	2	2	4	1	4	3
3	3	3	3	1	1	3
1	1	2	3	3	3	3

Poziția elementelor tabloului corespunde amplasării lor pe hartă. Două elemente din tablou se consideră vecine dacă regiunile corespunzătoare au o latură comună. De exemplu, fragmentul 4 4 va indica două localități vecine. Se prognozează ploi abundente. Pentru a evita

incidentele în urma inundațiilor posibile, în fiecare din localitățile(4) învecinate cu râuri(3) urmează să fie trimis câte un echipaj de salvatori. Se cere să se scrie un program care să determine câte echipaje de salvatori vor fi necesare (Olimpiada zonală între colegii. Zona Centru-Sud. 2020).

2. Safeu. Depozitul cu safeuri a băncii Gringotts Ltd are forma unui dreptunghi cu laturile $n \times m$, divizat în $n \times m$ platforme pătrate de dimensiunea 1×1 . Unica platformă liberă este în colțul din stânga sus. Ea marchează ieșirea din depozit. Unele dintre platforme sunt ocupate cu echipamente de supraveghere montate rigid, pe celelalte se află câte un safeu, care are dimensiunea platformei. În unul din safeuri se păstrează Piatra Filozofală. La cererea lui Dumbledore, spiridușii trebuie să scoată din depozit safeul cu Piatra Filozofală. Unica operație pe care o pot efectua ei este deplasarea unui safeu pe o platformă vecină, dacă aceasta este liberă. Se consideră vecine platformele care au o latură comună. Să se scrie un program care să determine numărul minim de operații necesare pentru a scoate safeul cu Piatra Filozofală din depozit. Safeul se consideră scos din depozit dacă este adus pe platforma de la intrare. (Concursul FII Campion. 2011, Iași).

Sunt frecvente situațiile, în care în calitate de elemente ale tablourilor 2D apar structuri. De exemplu, problemele în care se cere analiza, într-un careva mod, a elementelor tabelului periodic. În acest caz fiecare element al tabloului va descrie un element chimic, descrierea conținând denumirea elementului, masa lui atomică, informații privind structura atomului.

B1. Probleme celebre

Problemele celebre a căror enunțuri au legătură cu personalități marcante, evenimente istorice, fenomene naturale sau sociale, modele specifice de comportament uman sau animal au nu doar valoare cognitivă informatică – ele sunt surse de cunoștințe noi din diverse domenii, dar și un puternic motivator, în special la primele etape ale instruirii, când enunțurile problemelor sunt percepute mai mult ca texte descriptive decât modele abstracte.

Problemele celebre apar în diferite perioade istorice, ele au proprietatea de a reflecta și specificul social al epocii, care le-a generat. Un exemplu bun în acest context este problema ciclului Hamilton, cunoscută mai mult ca problema comisului – voiajor – una dintre problemele deschise pe grafuri. Problema, în limbaj cotidian, se formulează simplu: un comis-voiajor urmează să viziteze mai multe orașe, între unele dintre care există posibilitate de a călători. Condiția necesară este să le viziteze o singură dată. Fără această condiție problema este una foarte simplă, cu condiția aceasta – de nerezolvat în caz general. Dar de ce este impusă condiția? Răspunsul este unul literar, nu informatic – poate fi găsit la scriitorul american O Henry, care în

nuvela „The Gentle Grafter” descrie modul tipic de acțiune al comișilor voiajori în America începutului secolului XX.

Mai multe probleme matematice cu tentă algoritmică, relativ simple, din punct de vedere modern, au fost rezolvate în antichitate, iar rezolvările poartă și acum numele celor care le-au descris pentru prima dată.

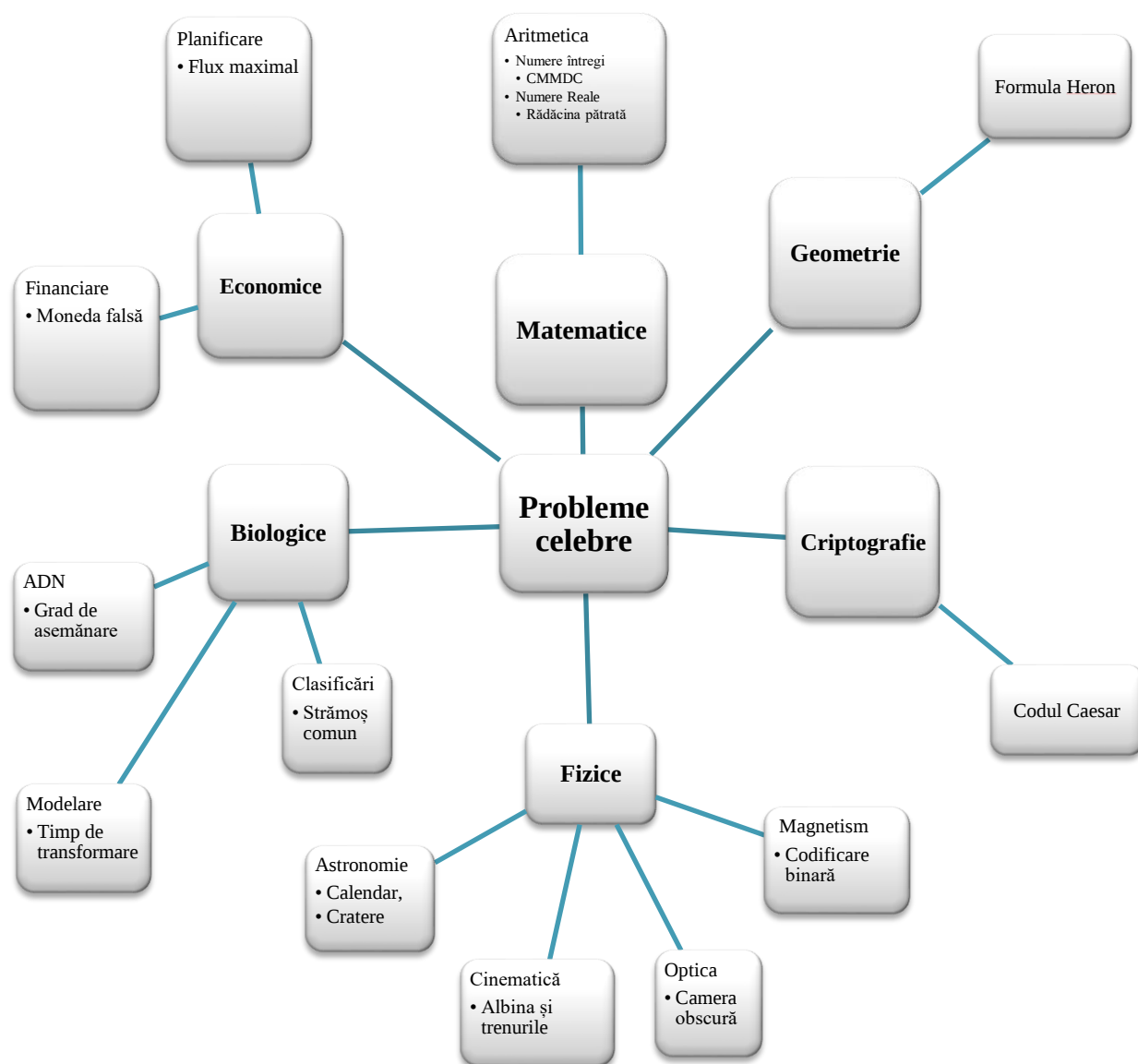


Figura 2.6. Principalele domenii de selecție a tematicii problemelor celebre propuse pentru rezolvare pe parcursul instruirii

Formula lui Heron (Hero). Este o formulă de calculare a ariei triunghiului după lungimile date a laturilor triunghiului. Din punct de vedere informatic are o importanță majoră, deoarece permite verificarea simplă a corectitudinii rezultatului, inclusiv prin calcule, efectuate

manual. Este un prototip potrivit pentru exemplificarea procesului de testare a datelor și soluțiilor. Formula este descrisă în numeroase ediții informatice și matematice [134], [84] [140].

Formula Babiloniană pentru calcularea rădăcinii pătrate. Una dintre cele mai vechi probleme matematice, cu soluție numerică, iterativă, care se reprezintă organic prin un algoritm. Rădăcina pătrată a unui număr S este așa un număr Z încât $Z \times Z = S$, Geometric, Z este lungimea laturii unui pătrat cu aria S .

Algoritmul are la bază câteva observații simple, și anume:

- Dacă x este o valoare care depășește numărul căutat z , ($x > z$), atunci $\frac{S}{x}$ va fi o valoare mai mică decât z , și invers: dacă $x < z$, atunci $\frac{S}{x} > z$.
- Media aritmetică $\frac{1}{2} \left(x + \frac{S}{x} \right)$ va fi o aproximare a rădăcinii pătrate mai bună decât x .
- Înlocuirea aproximării x cu o aproximare nouă $\frac{1}{2} \left(x + \frac{S}{x} \right)$ va îmbunătăți soluția calculată.
- Înlocuirea aproximării vechi cu aproximarea nouă, mai bună, poate fi efectuată repetat, atât timp cât îmbunătățirea soluției este relevantă.

În cazul implementării informatice problema poate fi folosită și pentru demonstrarea limitărilor sistemelor informatice, generate de reprezentarea numerelor în calculator. Pentru orice număr, diferit de pătratele perfecte, procesul de calcul a rădăcinii pătrate conform algoritmului nu se va sfârși niciodată. Modelul discret al mulțimilor numerice, implementat în informatică va duce la un rezultat diferit – după calcularea exactă a numărului maximal de cifre semnificative după virgulă, noile soluții vor coincide cu soluțiile precedente. Aceasta permite scrierea unui cod compact și intuitiv pentru algoritmul în studiu (s fiind numărul pentru care se calculează valoarea rădăcinii pătrate):

```
solutieNoua = 1;
do
{
    solutieVeche = solutieNoua;
    solutieNoua = (solutieVeche + s / solutieVeche) / 2;
}
while (solutieNoua != solutieVeche);
```

Problema celui mai mare divizor comun a două numere. Algoritmul lui Euclid. Euclid descrie soluția problemei în cartea sa „Elemente”. Traducerea soluției sună în felul următor: „Dacă două numere nu sunt prime între ele, scade numărul mai mic din cel mai mare și repetă procesul până când numerele devin egale. Acel număr este cel mai mare divizor comun” (sfârșit citat). Rezolvarea informatică este una simplă, cu atât mai simplă, cu cât mai avansate sunt

cunoștințele celor instruiți în limba engleză, deoarece transcrierea algoritmului într-un limbaj de programare tip C/C++ este foarte apropiată de descrierea în limbaj natural:

```
while(b != a)
{
    if (a > b) a = a - b;
    else b = b - a;
}
```

Problema prezintă interes și din punct de vedere al optimizării – algoritmul poate fi mai rapid dacă scăderile repetate sunt înlocuite cu calcularea restului întreg de la împărțire.

Sita lui Eratostene. Algoritmul, descris de Eratostene pentru a găsi numerele prime dintr-un interval. Descrierea originală a algoritmului s-a pierdut, dar este preluată și descrisă de Nicomachus în lucrarea „Introducere în aritmetică”: „*Pentru a găsi numerele prime, scrie toate numerele întregi de la 2 până la limita dorită. Apoi, ia primul număr (2) și elimină toți multiplii săi. Următorul număr rămas (3) este prim, iar apoi elimină toți multiplii săi. Continuă acest proces până când ajungi la un număr al cărui pătrat depășește limita. Numerele rămase sunt prime*” (sfârșit citat).

Implementarea informatică a algoritmului este simplă, intuitivă și se recomandă la etapa inițială a studierii tablourilor liniare. Poate fi folosită ca punct de start pentru identificarea primelor gemene, în contextul problemelor de concurs.

O altă categorie de probleme celebre sunt probleme de jocuri sau care se rezolvă pe o tablă de joc. Cele mai cunoscute în acest sens sunt problemele pe tabla de șah. Dintre acestea, cea mai cunoscută problemă, dar și cea mai simplă din punct de vedere a soluției este problema boabelor de orez.

Legenda spune că jocul de șah a fost inventat în India antică. Regele, auzind despre joc, a cerut să fie prezentat. I-a plăcut mult jocul și a vrut să răsplătească inventatorul, permițându-i să ceară orice recompensă. Inventatorul a cerut o recompensă aparent modestă: boabe de orez pe tabla de șah, după următoarea regulă: pe prima căsuță să primească 1 bob, pe a doua căsuță să primească 2 boabe, pe a treia căsuță să primească 4 boabe și așa mai departe, dublând numărul de boabe pe fiecare căsuță (citată după [83]).

Problema poate fi folosită pentru a demonstra limitele de valori pentru diverse tipuri de date informatice, motivarea implementării operațiilor cu numere mari, realizarea operațiilor pe biți pentru calcularea puterilor numărului 2 dar și în calitate de exercițiu pentru explorare.

Problema parcurgerii tablei de șah cu mișcări ale calului. O excelentă problemă de logică, care, la implementarea recursivă a reluării poate să ajute mult înțelegerea de către cei instruiți a mecanismului tehnicii de reluare, dar și a modelului recursiv de rezolvare a

problemelor informatice. Formularea problemei este foarte simplă: *se cere să se construiască un traseu pentru un cal pe tabla de șah, astfel încât calul să viziteze fiecare pătrat al tablei o singură dată.*

Majoritatea problemelor celebre provin din domeniul matematicii. Nu toate însă vin din antichitate. O problemă, care stă la baza înțelegerii noțiunii de graf, are originea în secolul XVIII (1736) este **problema podurilor din Königsberg**. Orașul este situat pe ambele maluri ale râului Pregel, dar și pe două insule, având multiple conexiuni (șapte poduri). Locuitorii orașului întreba dacă este posibil să faci o plimbare prin oraș, trecând o singură dată peste fiecare pod și revenind la punctul de plecare. Matematicianul Leonard Euler a găsit o interpretare schematică pentru problemă, a demonstrat lăsa soluției pentru configurația amplasării podurilor din Königsberg iar în baza acestei interpretări a fost dezvoltată ulterior noțiunea de *graf* [141], [142], [143].

Principiul fizic al propagării undei stă la baza **algoritmului Lee** (algoritmul undei). Este o metodă rapidă pentru determinarea unor domenii cu proprietăți date pe spații discrete. Se folosește în special pentru recunoașterea obiectelor, iar în scopuri didactice – pentru a rezolva problemele de identificare a celui mai scurt drum într-un labirint (doi sau trei dimensional). Este util și în calitate de simulator pentru studiul structurilor de date tip coadă.

Problema Turnurilor din Hanoi – o problemă perfectă pentru studiul complexității și calculării resurselor de timp. Legenda asociată problemei spune că *într-un templu din Hanoi există trei tije de diamant și 64 de discuri de aur de dimensiuni diferite, așezate de Buddha odată, demult, pe prima tijă în ordine descrescătoare, adică având cel mai mare disc la bază. Preoții templului au fost însărcinați să mute toate discurile de pe prima tijă pe a treia tijă, respectând următoarele reguli: 1. Doar un disc poate fi mutat la un moment dat. 2. Nu se permite plasarea unui disc mai mare peste unul mai mic. Profeția spunea că, atunci când toate discurile vor fi mutate, lumea se va sfârși.*

Problema este o ilustrare clasică a aplicării metodei *Divide et Impera* și a tehnicilor recursive de rezolvare. Suplimentar este un instrument foarte bun pentru cercetarea și estimarea complexității în funcție de dimensiunea datelor de intrare (n) pentru algoritmi exponențiali, [144] [107].

Problema rucsacului (Problema hoțului – the thief problem). O problemă cu componentă didactică pronunțată, care poate fi abordată pentru a ilustra principiul de rezolvare prin forță brută pentru dimensiuni relativ mici ale datelor de intrare, metoda relucrării – pentru seturi medii, metoda programării dinamice sau euristici tip Greedy – pentru dimensiuni mari ale datelor de intrare. În versiunea de problemă a hoțului, enunțul sună în felul următor: *într-un depozit este stocat un set din N articole, fiecare cu un volum și o valoare cunoscută. Un hoț pătrunde în*

depozit, având un rucsac de volum S . Determinați ce articole trebuie să selecteze hoțul, astfel încât volumul total al obiectelor selectate să fie mai mică sau egal cu S iar valoarea totală să fie cât mai mare posibilă.

Abordarea problemei pentru valori mic ($N < 20$) este posibilă chiar pentru începători. Baza soluției o va constitui algoritmul de generare a tuturor secvențelor binare de lungime N . De aici vine și limitarea – pentru $N = 20$, numărul secvențelor binare distincte va fi 1048576, ceea ce implică un timp de calcul, care va depăși 1 sec. pentru calculatoarele care pot efectua circa 1 000 000 operații / secundă.

În procesul de studiere a tehnicii reluării aceeași problemă poate fi soluționată până la valori ale lui N care sunt duble față de limitările forței brute – $N < 40$. Creșterea dimensiunii limită a problemei rezolvabile se datorează proprietății metodei reluării de a evita continuarea pe vectorii binari, despre care apriori se cunoaște că vor genera doar soluții inadmisibile.

La fel de elegantă este și abordarea dinamică a rezolvării. Limitările acesteia nu depind de valoarea N , ci de valoarea S . Limitările în acest caz sunt de obicei generate de memoria alocată, or, în funcție de formularea cerințelor problemei poate fi necesară alocarea unui câmp de $S \times N$ unități de memorie.

O abordare euristică, de multe ori nu suficient de exactă, dar foarte rapidă, este aplicarea tehnicii *Greedy*, care va propune o soluție suficient de bună într-un timp aproape liniar.

Mai multe probleme de această natură pot fi selectate din [44], [145], [86], [85], [112], [37], [119].

B2. Modele Matematice

Crearea modelului matematic este o etapă obligatorie în procesul de rezolvare informatică a problemelor matematice sau din alte domenii, reprezentând trecerea de la formularea inițială a problemei către algoritm, care, în mare măsură, este același model matematic, îmbogățit cu structuri adecvate de date, scopul cărora este micșorarea complexității sau, cel puțin, reducerea numărului de operații necesare pentru obținerea rezultatului.

Modelarea matematică și modelele matematice sunt deosebit de importante în cazul rezolvării unor probleme complexe. În funcție de restricții, modelul matematic poate fi mai simplu sau mai complicat pentru aceeași problemă, să permită o altă abordare informatică sau să accepte structuri de date diferite.

Primele modele matematice se aplică deja la etapa inițială de instruire, exemple clasice fiind formulele pentru calcularea rădăcinilor ecuațiilor pătratice, formula pentru calcularea iterativă a rădăcinii pătrate sau formula lui Heron pentru calcularea ariei triunghiului.

Odată cu creșterea complexității problemelor propuse pentru rezolvare crește și complexitatea modelelor matematice ale acestora. Curriculumul la disciplina Informatică prevede studierea modelelor matematice și a modelării matematice în clasa a XII-a, ceea ce este mult prea întârziat în cazul instruirii de performanță.

În contextul instruirii neformale primele modele matematice pot fi elaborate pornind de la primul nivel de instruire, la etapa instrumentală, analizând enunțuri cu model matematic descris explicit, cum ar fi problemele, în enunțul cărora figurează modele statice ale corpurilor geometrice, modele de mișcare a corpurilor fizice, sau ecuații ale reacțiilor chimice [35] [34] [146].

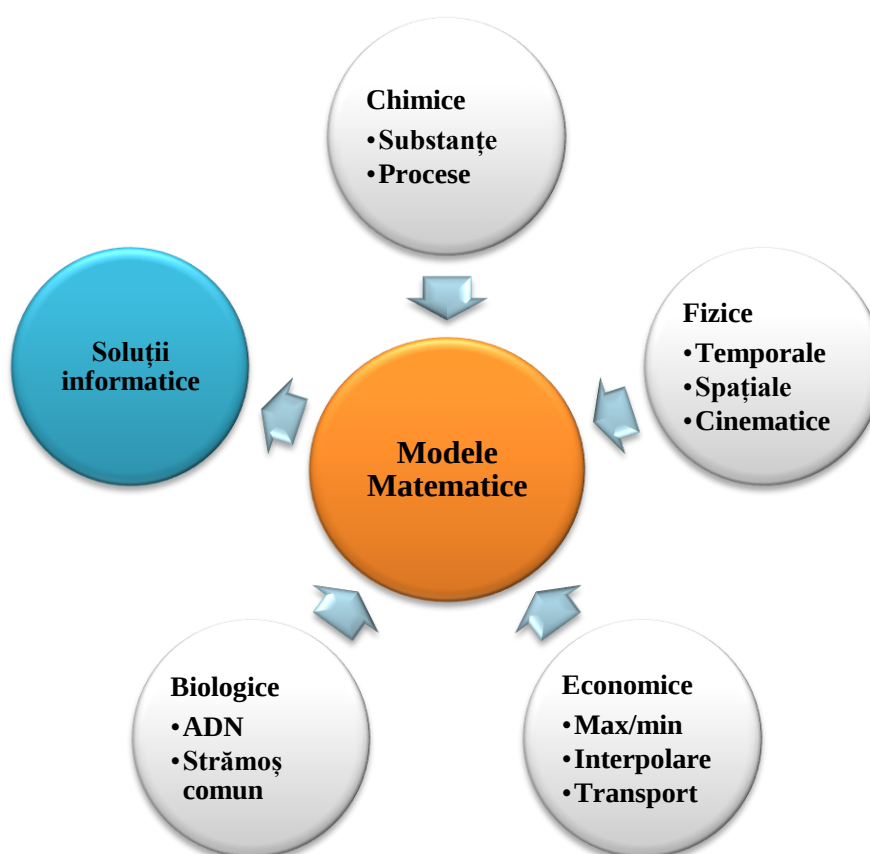


Figura 2.7. Procesul de rezolvare a problemelor din diverse domenii prin construcția modelelor matematice ale acestora

Exemplu: „în condiții de laborator, o populație de viruși, formată inițial din N unități și plasată într-un mediu steril, se micșorează în fiecare oră cu 50 de procente, dacă numărul virușilor la începutul orei este par, sau crește cu o unitate, dacă numărul virușilor la începutul orei este impar. În momentul când numărul virușilor devine mai mic decât cantitatea critică de supraviețuire C , populația dispăre integral. Să se scrie un program care va stabili timpul necesar, în ore, pentru distrugerea în laborator a unei populații din N viruși, având cantitatea critică de supraviețuire C ” [34].

Modelul matematic este descris explicit:

$$N_0 = N$$

$$N_i = \begin{cases} \frac{N_{i-1}}{2}, & \text{dacă } N_{i-1} \text{ este par} \\ N_{i-1} + 1, & \text{dacă } N_{i-1} \text{ este impar} \end{cases} \quad i = 1, 2, 3 \dots$$

În acest caz și modelul informatic este cât se poate de simplu:

```
t = 0;
while ( N >= C )
{
    t++;
    if ( N % 2 == 1 ) N++; else N = N / 2;
}
```

Problemele cu enunț geografic, unde se cere identificarea unui element sau a unei secvențe de elemente se reduc la un model matematic pe tablou bidimensional, în care elementele păstrează toate caracteristicile specifice condițiilor problemei. De exemplu, problema Defileu (Olimpiada zonală la informatică între colegii. Zona Centru. Anul de studii I, 2010) : *Ionuț este pasionat de călătorii cu bicicleta prin munți. El se pregătește pentru o excursie în munții Făgărașului și chiar a procurat o hartă a înălțimilor din zona în care va pleca. Harta reprezintă un tablou bidimensional $N \times N$, fiecare element al căruia indică înălțimea unui teren de suprafață unitară. O zonă liniară, continuă de teren (orientată orizontal) se numește defileu, dacă fiecare element al său are o valoare mai mică decât celelalte două elemente, amplasate în aceeași coloană a tabloului, cu o linie mai sus și una mai jos de acesta. Să se scrie un program care să determine cel mai lung defileu de pe hartă.*

Problema poate fi rezolvată prin o parcurgere consecutivă a liniilor tabloului, cu indicele (i) având valori de la 2 la (n - 1). Fiecare linie (i) este parcursă consecutiv, căutându-se coloane, care pot fi considerate începutul unui defileu. Odată identificată poziția de început, lungimea defileului crește cât timp elementul din linia (i), coloana (j) are o valoare mai mică decât elementele din aceeași coloană (j) din liniile (i-1) și (i+1). Cum numai această condiție nu se mai respectă, este semnalizat sfârșitul defileului. Lungimea defileului curent se compară cu soluția cea mai bună, identificată anterior și la necesitate o înlocuiește.

Modelele de rezolvare se pot modifica chiar în cadrul unei probleme, în funcție de etapa de rezolvare. În calitate de exemplu pentru următorul studiu de caz va fi propusă o problemă cu enunț din domeniul chimiei, aplicare reală militară, modelul necesar pentru obținerea soluției matematice fiind intuitiv unul fizic.

Coroziunea metalului (Federația Rusă, Olimpiada Națională la Informatică, 2000): „*pentru păstrarea a două tipuri de lichide agresive A și B se folosește un vas cu perete de separare cu straturi multiple, fabricat din N plăci de grosime diferită. Pentru fiecare placă i (i = 1, ..., N) este cunoscut timpul de dizolvare de către substanța A - a_i și substanța B - b_i.. Dizolvarea*

plăcilor are loc consecutiv, placă după placă, cu viteză constantă după grosimea stratului. Se cere proiectarea unui perete de separare, care ar rezista cât mai mult acțiunii substanțelor agresive.”

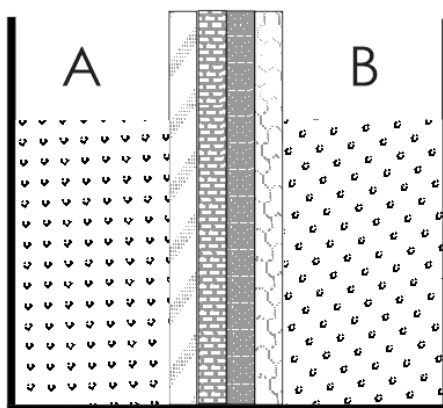


Figura 2.8. Ilustrație la enunțul problemei Coroziunea metalului

Modelul de rezolvare este destul de intuitiv: plăcile care sunt mai rezistente față de lichidul A sunt plasate cu atât mai în față spre A cu cât este mai mare raportul $\frac{a_i}{b_i}$. În realitate, acesta este modelul unui detonator chimic, care în funcție de ordinea montării plăcilor va întârzia mai mult sau mai puțin momentul contactului între lichide. Observația inițială, de a amplasa plăcile în ordinea descreșterii rezistenței față de lichidul A permite să se obțină timpul maximal de rezistență, dar nu și calcularea lui. Pentru a calcula timpul de rezistență, după ce plăcile sunt ordonate corect, se poate trece la un model fizic, cinematic, care descrie sistemul din două automobile A și B, care se mișcă în întâmpinarea unul altuia pe N porțiuni consecutive de drum. Automobilul A se află pe porțiunea de drum cu indicele st , automobilul B – pe porțiunea de drum cu indicele dr . Inițial $st = 1$, $dr = N$. Timpul necesar fiecărui automobil pentru parcurgerea segmentului de drum i este cunoscut. Cât timp automobilele se mișcă pe segmente diferite, rezolvarea se reduce la:

- determinarea segmentului x care se va consuma primul $x = \min(a_{st}, b_{dr})$;
- recalcularea timpurilor de parcurgere din ambele direcții pentru segmentul parcurs parțial: dacă $a_{st} > b_{dr}$, $dr - -$; $a_{st} = a_{st} - b_{dr}$; $b_{st} = \frac{(a_{st} - b_{dr})b_{st}}{a_{st}}$, în caz contrar, dacă $a_{st} < b_{dr}$, $st + +$; $b_{dr} = b_{dr} - a_{st}$; $a_{dr} = \frac{(b_{dr} - a_{st})a_{dr}}{b_{dr}}$.
- reluarea modelării mișcării pe segmentele de drum extreme

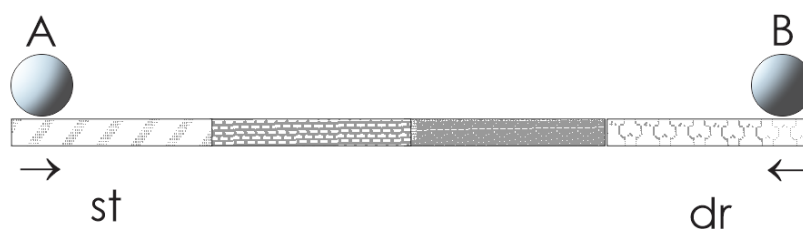


Figura 2.9. Modelul apropierei lichidelor la absorbția plăcilor extreme

La ultima etapă (când rămâne) o singură porțiune de drum se obține soluția poate fi reformulată astfel: Fie dat un segment de drum de lungime L , necunoscută. Automobilul A îl parcurge în timpul a , automobilul B – în timpul b . Peste cât timp se vor întâlni automobilele? Viteza A este $\frac{L}{a}$, viteza B este $\frac{L}{b}$. În acest caz viteza unui singur automobil virtual, care le combină pe ambele, va fi egală cu suma vitezelor automobilelor distincte, iar timpul necesar pentru acesta va fi $T = \frac{L}{\frac{L}{a} + \frac{L}{b}} = \frac{ab}{a+b}$ și nu va depinde de parametrul necunoscut L .

Problemele cu model matematic geometric sunt cel mai dificil de implementat. Modelele geometrice trebuie să prevadă toate situațiile, inclusiv cele extremale și încearcă să le separe în teste sau subprobleme separate.

Problemă-tip (Poveste, Olimpiada Balcanică, 2017) [147] În fiecare an, la mijlocul verii, un rege mic de statură cu numele Statu Palmă Barbă Cot, organizează în castelul său o petrecere, la care sunt invitați toți cavalerii din țările Balcanice. Făt Frumos este unul dintre invitați, și, ca un cavaler adevărat, vine la petrecere cu calul său năzdrăvan – Flămânzilă. Intrarea cu caii pe teritoriul castelului este interzisă, din acest motiv Făt Frumos intenționează să-l lege pe Flămânzilă de unul din copacii, care cresc în apropiere, în afara castelului. Flămânzilă este foarte cuminte atât timp cât are ce mânca (iarba lui preferată crește peste tot), dar, dacă a păscut toată iarba la care poate ajunge devine nervos și începe să sufle foc, ca un dragon. În acest moment Făt Frumos trebuie să părăsească petrecerea, pentru a liniști tovarășul său patruped. Pentru ca Flămânzilă să nu producă incendii, Făt Frumos trebuie să cunoască aria figurii în care se poate deplasa acesta. Figura este determinată de: copacul de care este legat calul (un punct de coordonate întregi X, Y), lungimea funiei L (un număr întreg, pozitiv), zidul castelului, peste care calul nu poate trece (Zidul formează un poligon convex cu N laturi)

Se cere să se scrie un program care va calcula aria figurii în interiorul căreia se poate deplasa Flămânzilă.

Modelul matematic al soluției este unul geometric și variază în funcție de valoarea parametrului L – lungimea funiei. Soluția problemei presupune analiza câtorva situații diferite în funcție de care se modifică setul de formule pentru calcularea rezultatului (Figura 2.8).

Cazul A: lungimea funiei este mai mică sau egală cu distanța de la copac (X,Y) până la poligonul care formează zidul. În acest caz aria solicitată se calculează ca aria cercului cu raza L.

Cazul B: lungimea funiei este mai mare decât distanța de la copac (X,Y) până la poligonul care formează zidul, dar nu depășește distanța până la vârfurile vizibile extreme ale poligonului. În acest caz se calculează coordonatele punctelor de intersecție a cercului cu poligonul P. Aria solicitată se calculează ca aria segmentului de cerc cu raza L și centrul (X, Y), mărginit de punctele a căror coordonate au fost calculate la pasul precedent, la care se adaugă aria poligonului simplu, determinat de centrul cercului, punctele de intersecție a cercului cu poligonul și vârfurile poligonului, situate în interiorul cercului.

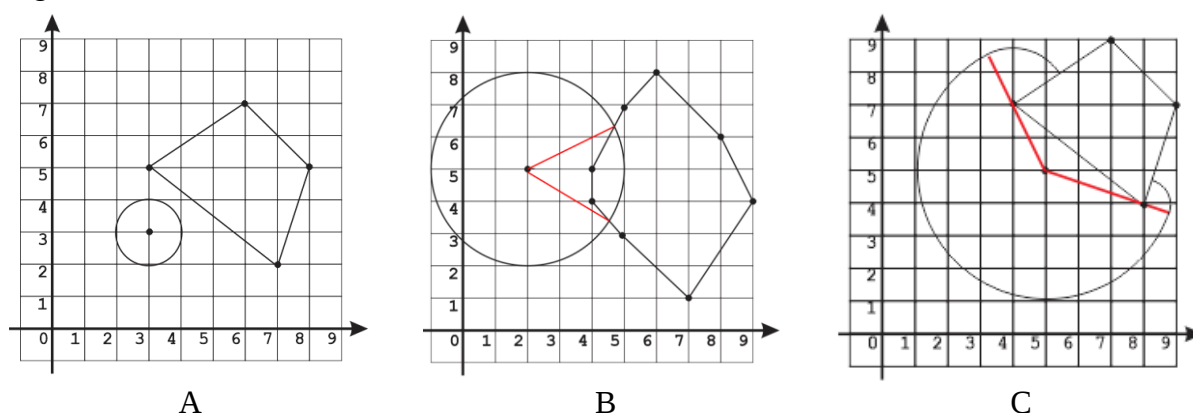


Figura 2.10. Modificarea modelului matematic și a algoritmului de rezolvare în funcție de subproblema rezolvată A – calcularea ariei unui cerc, B – calcularea ariei unui segment de cerc și a ariei unui poligon simplu, C – calcularea ariei unui poligon simplu și a multiplelor segmente de cerc

Cazul C: lungimea funiei depășește distanța până la vârfurile vizibile extreme ale poligonului. În acest caz se calculează aria segmentului de cerc cu raza L și centrul (X, Y), mărginit de vârfurile vizibile extreme ale poligonului, la care se adaugă aria poligonului simplu, determinat de centrul cercului și mulțimea de vârfuri vizibile ale poligonului, inclusiv cele extreme. La ultima etapă se modelează „îndoirea” funiei după laturi, fiind necesare calculele iterative a unor noi arii ale segmentelor de cerc.

Descrierea efectuată reprezintă doar primul nivel de abstractizare al modelului matematic al problemei. La următorul nivel se descriu formulele de calcul a tuturor mărimilor necesare pentru determinarea rezultatului final.

În continuare va fi descrisă detalierea pentru cazul A, care implică utilizarea unei singure formule de calcul – celei a ariei cercului:

$$S = \pi L^2$$

Calcularea soluției pentru cazurile B și C implică mai multe calcule intermediare, cum, de exemplu, determinarea coordonatelor punctelor (sau punctului) de intersecție a unui cerc cu un segment [140] [148] și obținerea soluției finale ca sumă de arii a segmentelor de cerc și poligoane simple.

Momentul crucial pentru modelele cazurilor B și C este determinarea coordonatelor punctelor de intersecție a laturilor unui poligon convex cu un cerc. Pentru latura cu numărul i , $i = 1, 2, 3, \dots, n$ se analizează trei cazuri distincte:

- latura nu are intersecție cu cercul cu centrul în punctul de coordonate date (X, Y) și raza L
- latura are o singură intersecție cu cercul cu centrul în punctul de coordonate date (X, Y) și raza L
- latura are două intersecții cu cercul cu centrul în punctul de coordonate date (X, Y)

Pentru determinarea situației care intervine la analiza laturii (i) se rezolvă sistemul de ecuații:

$$\begin{cases} Ax + By + C = 0 \\ (X - x)^2 + (Y - y)^2 - L^2 = 0 \end{cases}$$

Coeficienții A, B, C ai primei ecuații a sistemului se deduc din ecuația dreptei care trece prin două puncte, or, segmentul ce reprezintă latura poligonului furnizează toată informația necesară prin coordonatele extremităților sale.

În procesul de rezolvare a sistemului de ecuații se va ține cont și de faptul că unele intersecții ale cercului cu linia dreaptă se pot afla în afara segmentului în studiu. Prin urmare, sa va mai rezolva o subproblemă asociată soluției: determinarea apartenenței unui punct la un segment, toate punctele fiind descrise prin coordonatele lor carteziane.

Modelul informatic asociat este unul mult mai complex. O realizare funcțională posibilă este prezentată în Anexa 1.

B3. Elemente de complexitate

Capacitatea de analiză a complexității problemelor în funcție de resursele necesare de timp și memorie permite să fie estimate corect limitările metodelor utilizate, timpul necesar pentru obținerea rezultatului dimensiunile limită ale datelor, care pot fi prelucrate. Studiul bazelor complexității este planificat de Curriculumul la disciplină ca un modul pentru predare / învățare în clasa a XII-a, ceea ce este mult prea târziu pentru cei care participă activ la diverse concursuri de programare.

Manualul școlar [34] conține o descriere calitativă a componentelor teoretice ale complexității, prin urmare poate fi folosit în calitate de resursă de învățare principală la această ultimă etapă a nivelului intermediar de pregătire, indiferent de vârsta celui instruit. Totuși, în contextul specificului spațiului de probleme, care se rezolvă la acest moment, este necesară introducerea unor noțiuni și clasificări specifice, cum ar fi de exemplu clasele de probleme, reductibilitatea problemelor, complexitatea operațiilor pe structuri de date, dar și structurile de date specifice, pe care sunt definite mai multe probleme abstracte și modele economice, sociale, militare reale.

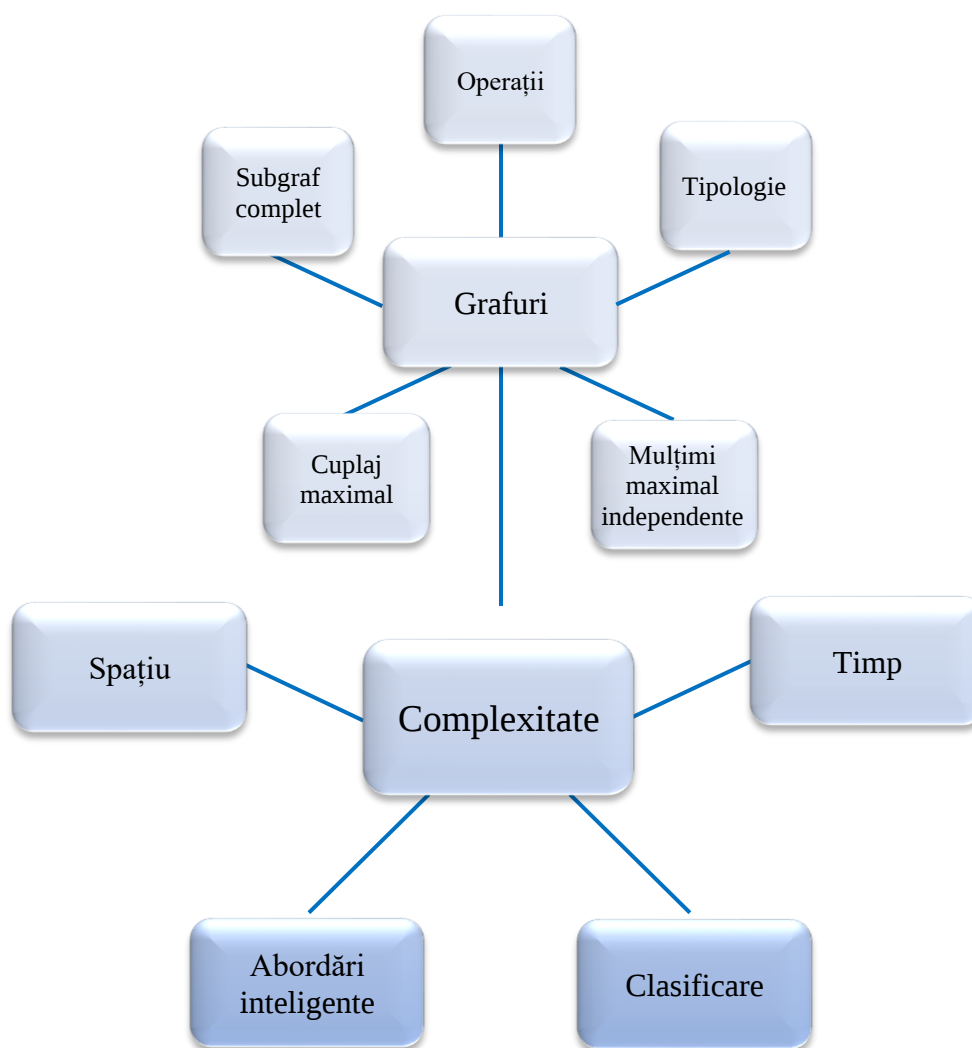


Figura 2.11. Tipuri de complexitate și complexitatea structurilor de date

Primele noțiuni cu care urmează a fi completate cunoștințele din domeniul complexității sunt funcțiile de complexitate asimptotice și notațiile complexității:

- Notăția O (Big - O) – reprezintă limita superioară a timpului de rulare a unui algoritm. Ea oferă cea mai rea estimare pentru complexitatea algoritmului. Grafic, ea reprezintă o funcție $g(n)$, care, începând cu un careva n_0 , pentru o constantă C pozitivă, satisface relația $f(n) \leq Cg(n)$.

- Notăția Ω (Big - Ω) – reprezintă limita inferioară a timpului de rulare a unui algoritm. Ea oferă estimarea timpului minim garantat de funcționare a algoritmului. Grafic, ea reprezintă o funcție $g(n)$, care, începând cu un careva n_0 , pentru o constantă C pozitivă, satisface relația $f(n) \geq Cg(n)$.
- Notăția Θ (Big - Θ) – reprezintă „canalul” de complexitate a unui algoritm. Ea oferă estimarea diferenței între timpul maximal de funcționare a algoritmului pentru orice valoare $n > n_0$ și timpul minim garantat. Grafic, ea reprezintă o funcție $g(n)$, care, începând cu un careva n_0 , pentru constantele pozitive C_1, C_2 $C_1 < C_2$, satisface relația $C_1g(n) \leq f(n) \leq C_2g(n)$.

În condițiile unei competiții cea mai relevantă este notația O , numită în continuare funcția de complexitate.

Practic, noțiunile pot fi introduse prin analiza comparativă a metodelor de sortare eficiente. Se recomandă studierea și implementarea algoritmilor eficienți de sortare QuickSort, MergeSort [24], [149] și lansarea repetată a acestora pe seturi mari de date, cu măsurarea timpului de funcționare pe fiecare set de date și contorizarea „victoriilor” fiecăruia dintre algoritmi. Demonstrația poate fi începută cu participarea și a unuia din algoritmi clasici de sortare BubbleSort sau sortarea prin Selecție / Inserție. Acesta va urma să fie exclus din competiția algoritmilor după lansarea testelor cu tablouri a căror număr de elemente depășește ($n = 50\,000$).

Prin calcule elementare se determină complexitatea fiecărui algoritm [150] [151] - $O(n^2)$ pentru metodele clasice, $O(n \times \log_2(n))$ – pentru MergeSort și aceeași complexitate pentru seturile aleatorii de date și în cazul QuickSort. Totuși, urmează să fie generate și seturi de date, care să demonstreze instabilitatea complexității acestui din urmă algoritm – pentru seturile de date, care dezechilibrează arborele de sortare al algoritmului QuickSort eficiența acestuia scade până la $O(n^2)$.

Limitările funcțiilor de complexitate exponențiale pot fi demonstrate în cel mai simplu mod prin generarea vectorilor binari de lungime N – se cunoaște că numărul acestora este 2^N . Finalizarea problemei devine tot mai lentă odată cu creșterea N . Tabelul de observație generat prezintă, cu devieri nesemnificative, în funcție de performanța sistemului informatic utilizat, același model de creștere a timpului necesar (Tabelul 2.2):

**Tabelul 2.2. Demonstrația creșterii timpului de execuție
în cazul algoritmilor exponențiali**

N	4	5	10	15	20	25	30	35	40
T(N) sec	0.029	0.04	0.05	0.06	0.065	0.16	3.6	114	--

Limitările de spațiu sunt generate, de obicei de consumul mare de memorie în cazul utilizării programării dinamice, metodă, care impune păstrarea stărilor odată calculate pentru a evita calculele repetate. Consumul excesiv de memorie se atestă și în cazul algoritmului de sortare MergeSort, care folosește un tablou auxiliar de dimensiunea tabloului inițial pentru reordonarea fragmentelor.

Clase de probleme. După stabilirea diferenței de eficiență a algoritmilor, a căror funcție de complexitate este descrisă de un polinom, și a algoritmilor, a căror funcție de complexitate este una exponențială, se poate discuta clasificarea problemelor informatice, fiind introduse clasele P, NP, și NP-complete cu exemple de probleme pentru fiecare dintre ele.

Pentru problemele din clasa P – probleme a căror algoritmi de rezolvare au complexitate polinomială, este importantă posibilitatea reducerii complexității cu o unitate de putere N, prin înlocuirea structurilor de date liniare cu structuri neliniare. Demonstrația practică – la studierea structurilor de date neliniare. Utilizarea arborilor tip Heap permite obținerea unui algoritm de sortare de complexitate $O(n \times \log_2(n))$, aceleași structuri Heap în cazul algoritmului Dijkstra permit micșorarea timpului de calcul de la $O(n^2)$ până $O(n \times \log_2(n))$.

Clasa NP este clasa de probleme, pentru care nu se cunosc la moment algoritmi eficienți de rezolvare. Este demonstrat însă, că dacă se obține un algoritm polinomial de rezolvare pentru una dintre problemele clasei NP, atunci toată clasa de probleme se dovedește a fi rezolvabilă în timp polinomial [31] [68].

Analiza problemelor din clasa NP este importantă în contextul propunerilor frecvente a problemelor din această clasă la competițiile de programare, dar și a caracterului multidisciplinar al structurilor de date în studiu. Grafurile reprezintă nu numai structuri matematice, ele pot fi privite ca modele a rețelelor de orice natură, care permit transmiterea semnalelor între noduri și crearea asocierilor funcționale. Prin urmare, se recomandă realizarea acestui studiu în paralel cu introducerea grafurilor în calitate de structuri de date abstracte și a algoritmilor asociați grafurilor, cu studiul tehnicilor de programare și cu studiul algoritmilor genetici.

Grafuri. Definițiile, operațiile și algoritmii de bază pe grafuri sunt studiate suficient de bine, [141], [28], [90], [27], iar metodologia de predare în cele mai diferite contexte este un element fundamental în orice centru de performanță informatică [152], [143], [142].

La etapa de studiu al complexității vor fi scoase în evidență problemele din clasa NP, formulate în termenii Teoriei Grafurilor:

- Problema mulțimilor maximal independente
- Problema cuplajului maximal

- Problema subgrafului complet de grad maximal

Tradițional, studiul începe cu problema Mulțimilor Maximal Independente (MMI), după care se introduc noțiunile de *graf complementar* și *graf asociat* și se demonstrează, prin transformări polinomiale că soluția problemei MMI poate fi folosită în calitate de soluție a problemei cuplajului maximal în graful asociat și soluție a problemei subgrafului complet de putere maximală în graful complementar.

Minimul necesar pentru lansarea rezolvărilor este definirea corectă a noțiunilor asociate MMI, un exemplu relevant și o formulare de problemă captivantă:

Fie dat un graf neorientat $G = (V, E)$.

Definiții:

- *Mulțime independentă* se numește o mulțime de vârfuri ale grafului, astfel încât oricare două vârfuri din ea nu sunt unite direct prin muchie. Formulată matematic, mulțimea independentă S este o mulțime, care satisface relația: $S \subseteq V : S \cap \Gamma(S) = \emptyset$.
- Mulțimea S se numește *maximal independentă*, dacă nu există o altă mulțime independentă S' , pentru care se îndeplinește condiția: $S \subseteq S'$.

Exemplu:

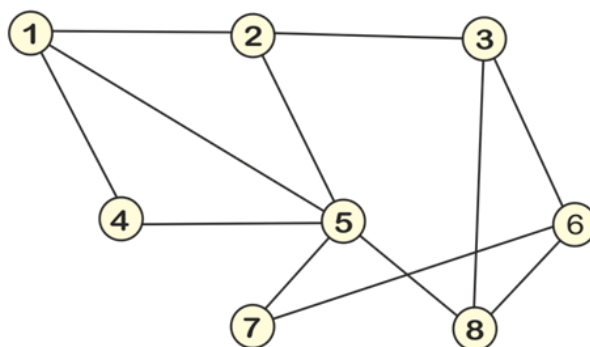


Figura 2.12. Mulțimile vârfurilor $\{1, 3, 7\}$ $\{2, 8, 7, 4\}$ $\{4, 6\}$ – mulțimi independente.

Mulțimile $\{2, 8, 7, 4\}$, $\{2, 4, 6\}$ sunt maximal independente, iar $\{1, 3\}$, $\{2, 4\}$ – nu

Probleme-tip: 1. (Paza regală, propusă CEOI 2002) Odată demult era un regat. Era acolo tot ce-i trebuie unui regat, chiar și rege cu castel. Planul castelului era un dreptunghi, împărțit în $M \times N$ pătrate unitare. Unele pătrate erau pereți, altele libere. Fiecare pătrat liber e numit odaie. Regele, fiind paranoic, a pus în unele odăi fântâni ascunse (cu aligatori la fund).

Apoi a decis să se pună peste tot unde era posibil în castel străji. Ceea ce nu era de loc simplu. Străjile sunt antrenate să tragă imediat ce văd pe cineva, chiar pe altă strajă. Regele trebuie să plaseze străjile astfel, încât ele să nu se vadă – altfel se împușcă reciproc! Suplimentar, ei nu pot fi plasați în odăile cu fântâni. Fiecare odaie are nu mai mult de un străjer. Doi

străjeri din odăi diferite se văd reciproc, dacă odăile sunt în aceeași linie sau coloană și între ele nu sunt pereți

2. (Olimpiada zonală la informatică între colegii. Zona Centru. Anul de studii III, propusă 2014) *Pentru a asigura securitatea turiștilor în zona de munte, s-a decis instalarea punctelor de observare a salvamontiștilor pe unele vârfuri ale munților Făgărașului. Un punct de observare controlează vârful muntelui pe care este amplasat, dar și toate vârfurile munților vizibili din acesta. Punctele de observare nu pot fi vizibile între ele – pot să apară confuzii în identificarea grupelor de turiști. Noul șef al serviciului de observare dorește să țină sub observație toate vârfurile munților din regiune, amplasând un număr cât mai mic de puncte de observare. Se cere să se scrie un program care să determine numărul minim de puncte de observare necesare pentru a controla toate vârfurile munților din zonă. punctele de observare vor fi invizibile între ele.*

C1 – C3. Tehnici – Structuri - Metode netradiționale de rezolvare a problemelor

Etaple nivelului avansat de instruire nu pot fi cercetate și realizate separat. Tehnicile de programare presupun utilizarea unor structuri specifice de date, la fel utilizarea algoritmilor genetici, mai puțin tradiționali. Algoritmii clasici cu autoinstruire [87] au nevoie de structuri de date adecvate. Algoritmul jocului Viața preia elemente ale programării dinamice, iar cunoașterea mecanismelor de estimare a complexității – absolut necesară.

Tehnicile de programare sunt metode generale de abordare a rezolvării problemelor, pentru care nu sunt cunoscuți și demonstrați algoritmi eficienți. În contextul discuțiilor despre complexitate se definesc și tehnicile de programare mai puțin cunoscute. Specificul acestei etape este nu atât studierea tehnicilor de programare propriu-zise – către această perioadă cei instruiți sunt deja cunoscuți cu ele – cât utilizarea lor creativă, în funcție de problemă.

Procesul de instruire pe etapa C1 începe de la o scurtă repetare / prezentare a tehnicilor de programare, cu rezolvări de probleme-tip, selectate pentru a pune în evidență specificul metodei (Figura 2.13).

Metoda trierii este una din tehnicile generale de rezolvare a problemelor care, în varianta sa clasică nu se supune optimizării. În linii mari, metoda trierii se aplică fie în cazul problemelor deschise, pentru care nu există algoritmi de rezolvare optimi de tip polinomial sau exponențial, care admit excluderea unor stări ale sistemului cercetat. Este aplicată larg pentru problemele de criptografie (spargerea parolelor prin generare consecutivă) dar și pentru demonstrații a afirmațiilor matematice – prin analiza (calcularea) tuturor cazurilor posibile enunțate.

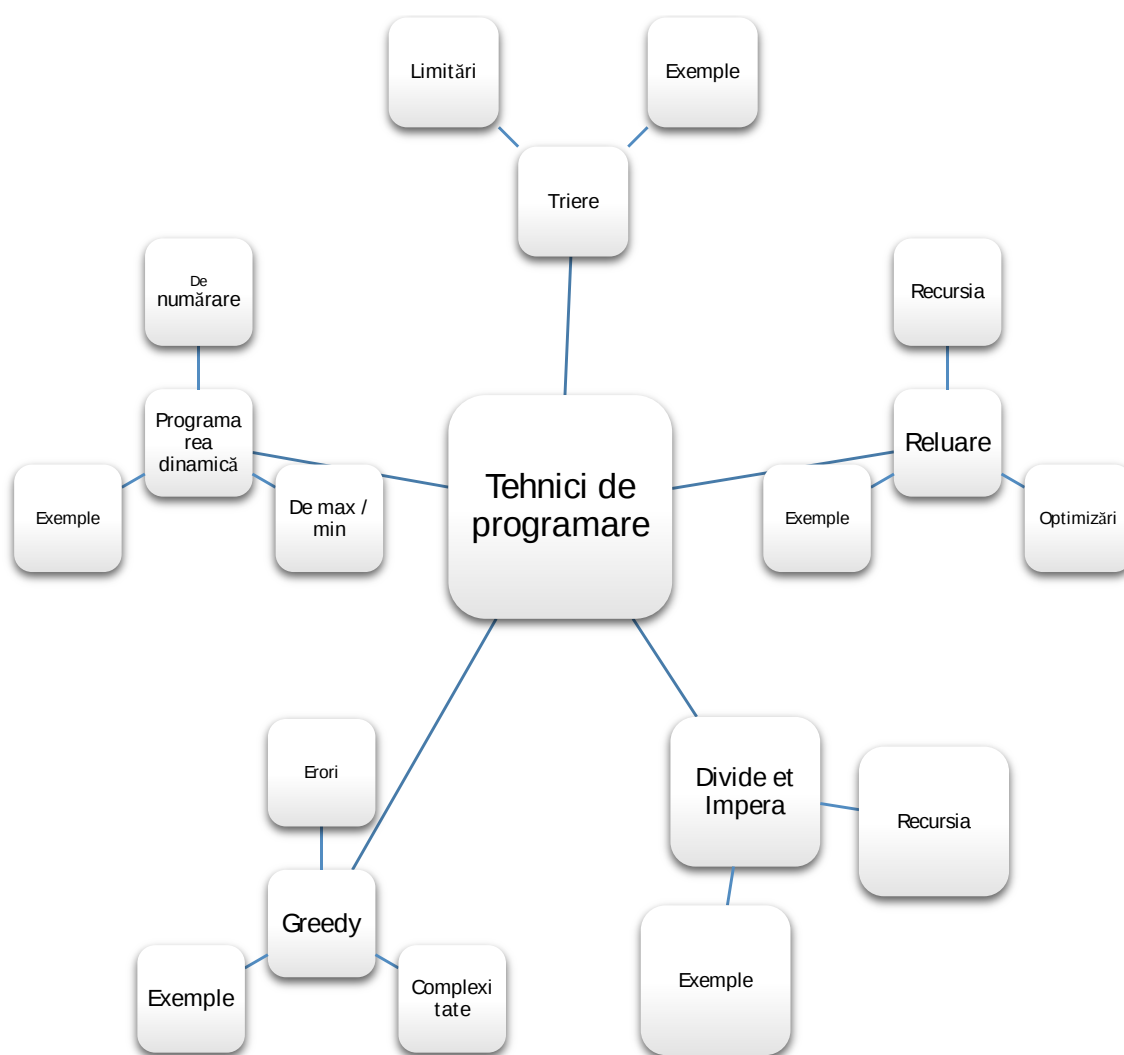


Figura 2.13. Puncte de reper de studiere a tehnicilor de programare

Metodologia de demonstrare a unei afirmații (sau de obținere a celei mai bune soluții) se bazează pe două componente:

- Demonstrarea numărului finit de stări posibile ale sistemului (sau generarea tuturor stărilor posibile ale sistemului cercetat – în funcție de natura problemei)
- Verificarea fiecărei stări a sistemului și demonstrarea că aceasta este sau nu este soluția parțială a problemei în proces de rezolvare. Soluția generală este determinată fie în paralel, fie ulterior ca fiind cea mai bună dintre soluțiile parțiale.

Utilizarea metodei pentru identificarea parolelor – una din aplicațiile de bază a metodei trierii. Considerând în calitate de generator de parole un dispozitiv cu productivitatea de 10^6 combinații de simboluri a unui alfabet clasic de 36 simboluri³ (în realitate alfabetele utilizate pentru generarea parolelor pot avea până la 224 simboluri) se va încerca determinarea unei

³ 26 simboluri ale alfabetului latin și 10 simboluri numerice.

lungimi „sigure” pentru o parolă aleatorie. Parola este considerată sigură, dacă pentru spargerea ei prin generarea tuturor secvențelor posibile ale alfabetului este necesar un timp care depășește un an.

**Tabelul 2.3. Dependența timpului necesar
pentru determinarea unei parole prin forță brută**

Lungimea parolei	Numărul de combinații	Rezistența ⁴	Timpul necesar
1	36	5	Mai puțin de 1 sec
2	1296	10	Mai puțin de 1 sec
3	46 656	15	Mai puțin de 1 sec
4	1 679 616	21	Mai puțin de 2 sec
5	60 466 176	26	1 min
6	2 176 782 336	31	36 min
7	78 364 164 096	36	Apr. 22 ore
8	2,821 109 9x10 ¹²	41	Apr 32 zile
9	1,015 599 5x10 ¹⁴	46	Apr. 3,2 ani
10	3,656 158 4x10 ¹⁵	52	Apr. 115 ani
11	1,316 217 0x10 ¹⁷	58	Apr. 4174 ani
12	4,738 381 3x10 ¹⁸	62	Apr. 150 000 ani

Astfel, se poate deduce concluzia că o parolă sigură la atacurile de forță brută trebuie să aibă o lungime de cel puțin 9 simboluri.

Exemplul propus ilustrează cât se poate de bine ineficiența metodei trierii, ca metodă practică de programare. Chiar și în cele mai simple probleme cu alfabet de două simboluri posibile (probleme discrete 0 : 1) creșterea timpului necesar pentru generarea secvențelor binare (fără analiza soluțiilor parțiale, care le corespund) este descrisă asimptotic⁵ de tabelul 2.3.

Tabelul 2.4. Timpul necesar pentru generarea secvențelor binare de lungime N

Lungimea secvenței, N	Număr de combinații	Timp de generare	Lungimea secvenței, N	Număr de combinații	Timp de generare
5	32	< 1 sec	40	2 ⁴⁰	Apr. 5 ore
10	1024	< 1 sec	50	2 ⁵⁰	Apr. 213 zile
15	32768	< 1 sec	60	2 ⁶⁰	597 ani
20	1048576	< 1 sec	70	2 ⁷⁰	611 328 ani
25	33554432	< 1 sec	80	2 ⁸⁰	626 000 mii ani
30	1073741824	20 sec	90	2 ⁹⁰	641 022 milioane ani
35	34359738368	Apr. 11 min	100	2 ¹⁰⁰	656 406 miliarde ani

Codul realizat pentru generarea secvențelor binare exclude toate operațiile auxiliare și tipurile de date ineficiente, prin urmare timpul de execuție nu mai poate fi micșorat pentru sistemul de calcul utilizat:

⁴ Rezistența parolei este o caracteristică numerică calculată ca logaritmul în baza doi de la numărul de parole posibile, rotunjit prin trunchiere

⁵ Pentru un sistem de calcul cu 4 nuclee Atom D520, 1,5 MHz

Ultimul exemplu permite să stabilim limitele rezonabile a dimensiunii problemei pentru un sistem monoprosesor, fără paralelizare de procese: maxim 40 elemente.

Metoda reluării (backtracking), definită și elaborată în procesul dezvoltării Inteligenței Artificiale, este rezultatul unei istorii din 1852, când studentul englez Francis Guthner a enunțat problema celor 4 culori: sunt suficiente 4 culori pentru a colora o hartă ce reprezintă diverse țări, cu condiția ca oricare două țări vecine (cu frontiera comună) să fie colorate cu culori diferite [153]. Teoretic, problema a fost rezolvată pentru 5 culori, dar rezolvarea ei completă - pentru 4 culori ($n > 4$) a fost posibilă doar în 1977 (K. Appel și V. Haken) prin utilizarea calculatorului.

În general metoda se aplica pentru problemele ale căror soluții se pot reprezenta sub formă de vectori $S = (x_1, x_2, \dots, x_n) = \prod_{i=1}^n S_i$ unde:

S - spațiul soluțiilor posibile, S_i - elementele acestui spațiu.

Complexitatea problemei este determinată de dimensiunea spațiilor soluțiilor posibile.

Problemele tip pentru rezolvare prin metoda reluării sunt:

- *problemele de descompunere: se dă un număr natural N , să se obțină toate descompunerile acestuia în termeni naturali (cu diverse variații);*
- *probleme pe tabla de șah: parcurgerea tablei cu mișcări ale calului, amplasarea a N dame;*
- *Probleme în labirint: numărarea tuturor traseelor cu diverse proprietăți*

Abordarea recomandată pentru utilizarea acestei metode este recursia. Soluțiile recursive de reluare sunt compacte ușor de înțeles și de testat.

Divide et Impera. O metodă eficientă pentru rezolvarea problemelor complexe prin reducerea dimensiunii repetată a acestora până la cazuri elementare și restabilirea soluției finale prin integrarea soluțiilor obținute la etapele precedente [154] [155] [138]. Fiind universală, este descrisă exhaustiv, reprezentante tipice ale metodei fiind problemele de frontieră cu tentă matematică și informatică Turnurile din Hanoi [83], sortarea MergeSort [24], cea mai apropiată pereche de puncte [31].

Din schema generală a metodei (Figura 2.14) se observă că procesul de rezolvare are loc doar la etapa divizării finale, în rest se are grijă doar de asamblarea corectă a soluției.

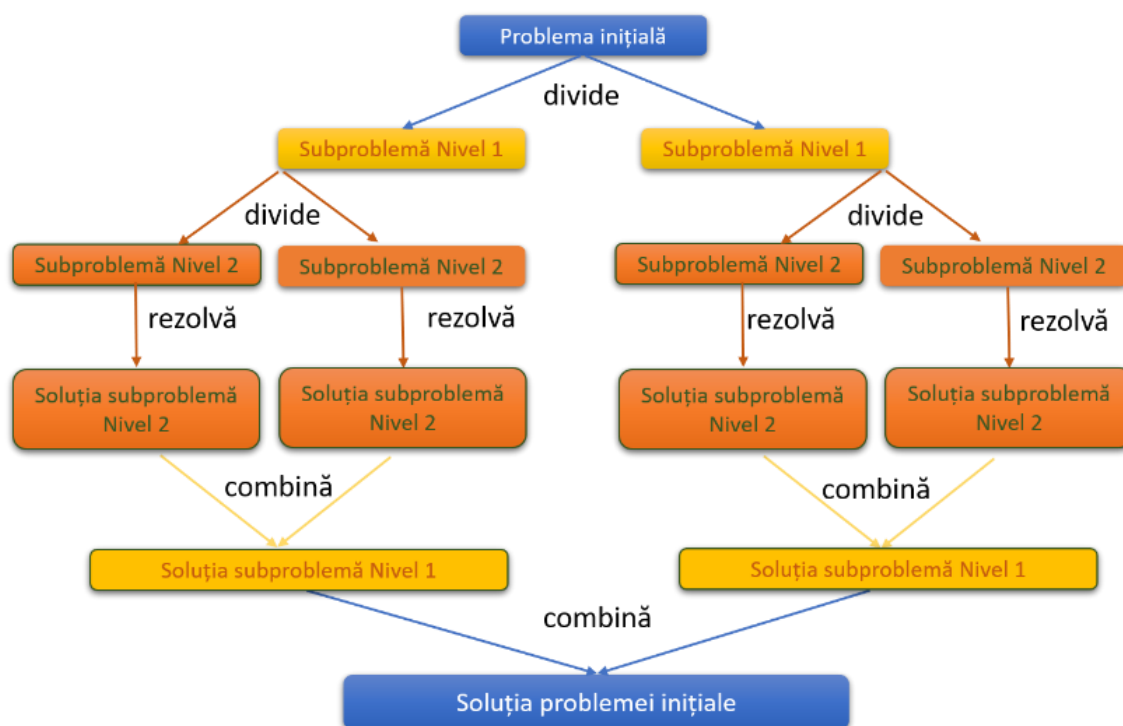
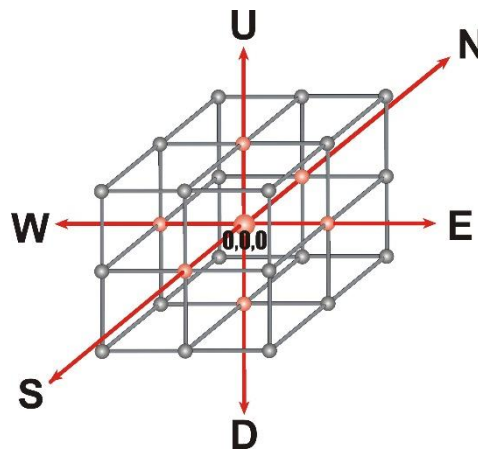


Figura 2.14. Ilustrarea principiilor de rezolvare a problemelor prin tehnica Divide et Impera

Programarea dinamică este considerată cea mai eficientă tehnică de programare. Esența metodei este obținerea soluției pe etape, folosind la etapa (sau etapele) următoare soluțiile obținute anterior. Suplimentar, se evită recalcularea acelorași valori, cercetarea repetată a situațiilor intermediare, ceea ce permite optimizarea esențială a timpului de lucru pentru algoritmi dinamici. Teoria programării dinamice este introdusă și dezvoltată de Richard Bellman [156] [157] [158], iar componentele aplicative sunt cercetate și introduse de Dijkstra [159], [160]. Specificul utilizării tehnicii de programare dinamică în concursuri este descris în [144], [161], [162].

Problemele-tip pentru introducerea în metoda programării dinamice se consideră generarea sau *calcularea numerelor Fibonacci*, cu demonstrarea ineficienței abordării recursive, și soluția dinamică pentru *problema rucsacului*. În calitate de studiu de caz se poate propune analiza comparativă a soluțiilor, obținute prin diferite metode.

O problemă mai puțin obișnuită a fost propusă la Olimpiada de Informatică între colegii, zona Centru-Sud 2010: *NASA a elaborat un nou model de robot pentru lucrări în modulele stațiilor orbitale. Modulele stației formează o rețea cubică, fiind unite între ele prin tunele orizontale și verticale de lungime unitară. Algoritmul de deplasare a robotului este format din instrucțiuni, care îl deplasează cu o unitate spațială spre Nord(N), Sud(S), Vest(W), Est(E), Sus(U) sau Jos(D). În algoritm instrucțiunile se pot repeta. După sfârșitul îndeplinirii algoritmului robotul se oprește.*



Axele de coordonate ale stației orbitale sunt paralele direcțiilor geografice, conform desenului. Prin urmare, modulele sunt plasate în puncte cu coordonate întregi. Unitatea de măsură coincide cu lungimea unui tunel, care unește două module vecine. Inițial robotul se află în modulul cu coordonatele $(0,0,0)$, amplasat în centrul stației orbitale. Să se determine numărul total de algoritmi distincți din exact K instrucțiuni care deplasează robotul din modulul cu coordonatele $(0,0,0)$ în modulul cu coordonatele (X, Y, Z) .

Etapă de studiere a tehnicilor de programare poate fi separată în sens de studiere a conținuturilor, metodelor specifice de aplicare, mini cercetărilor ghidate, efectuate de cei instruiți, de exemplu: [163], [164], [165].

Structurile de date complexe. La fel ca și grafurile sunt structuri neliniare, dar, de obicei reprezintă arbori binari cu noduri grupate pe nivele, cu diverse proprietăți suplimentare (Figura 2.15). Studiarea acestor structuri începe odată cu începutul rezolvării problemelor interactive, sau a jocurilor cu funcții ale sistemului, în special în cazurile când urmează să se răspundă la un număr cât mai mic de interogări.

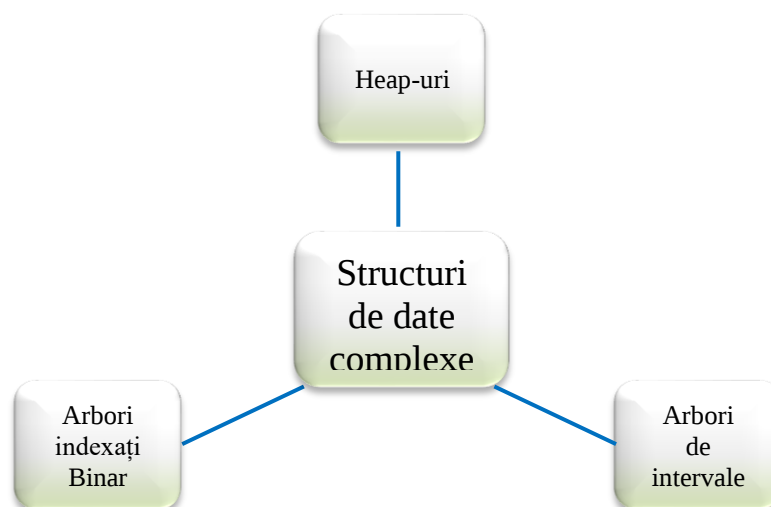


Figura 2.15. Structuri de date complexe

Arborii binari de căutare (ABC) cele mai populare structuri neliniare, care permit căutarea / identificarea rapidă a elementelor unei mulțimi, în funcție de valoarea unei chei. Tradițional, , asupra cheilor nodurilor se impun câteva proprietăți suplimentare:

- fiecare nod al arborelui binar este mai mare decât fiul lui stâng (nodurile din subarborele stâng)
- fiecare nod al arborelui binar este mai mic decât fiul drept (nodurile din subarborele drept).

Arborii cu asemenea proprietăți se numesc *arbori de căutare – sortare*, sau, mai des – *arbori binari de căutare*. Ei pot fi utilizați eficient pentru căutare sau ordonarea prin o simplă parcurgere a informației indexate după cheile respective. Problemele-tip de concurs pentru utilizarea arborilor binari de căutare sunt problemele de criptografie [91], [92] sau problemele interactive [104], [71].

Heap-urile. Cele mai recomandate structuri neliniare pentru instruire. În heap-uri este folosită în mod explicit legătura dintre arborii binari și tablourile liniare, demonstrându-se modelul de organizare a structurilor neliniare (arbori binari compleți) în formă de tablouri liniare:

Heap binar – un Heap organizat în baza unui arbore binar, astfel încât fiecare nod (element), are cel mult doi fii (două noduri din nivelul imediat următor, conectate la el). Heap-ul binar are următoarele proprietăți:

- Este un arbore binar „aproape complet” (nodurile se adaugă consecutiv – de la stânga spre dreapta (pe nivel), de sus în jos (între nivele))
- Dacă nodul X este părinte pentru Y, Z, atunci valoarea lui X este mai mare decât oricare dintre valorile Y, Z (max-Heap) / mai mică decât oricare dintre valorile Y, Z (min-Heap)

Chiar dacă heap-urile sunt descrise ca structuri arborescente, modelul consecutiv de adăugare a nodurilor noi în heap permite ca acestea să fie reprezentate prin structuri de date mai puțin complexe și mai ușor de manipulat – vectori, sau, pentru cei care folosesc structurile de date clasice – tablouri (array).

Dacă nodurile unui heap se numerează consecutiv de la rădăcină, de la stânga spre dreapta, de sus în jos, începând cu indicele 1 pentru rădăcină, se observă următoarea legitate: dacă nodul „părinte” are indicele $[i]$, atunci fiii lui vor avea indicii $[2i]$ și $[2i + 1]$.

În cazul în care numerotarea începe de la 0 (ceea ce este caracteristic tablourilor în limbajele de programare C / C++), legătura între indicii nodului - părinte și indicii nodurilor – fii este:

$$[i] \rightarrow [2i + 1] \text{ și } [2(i+1)].$$

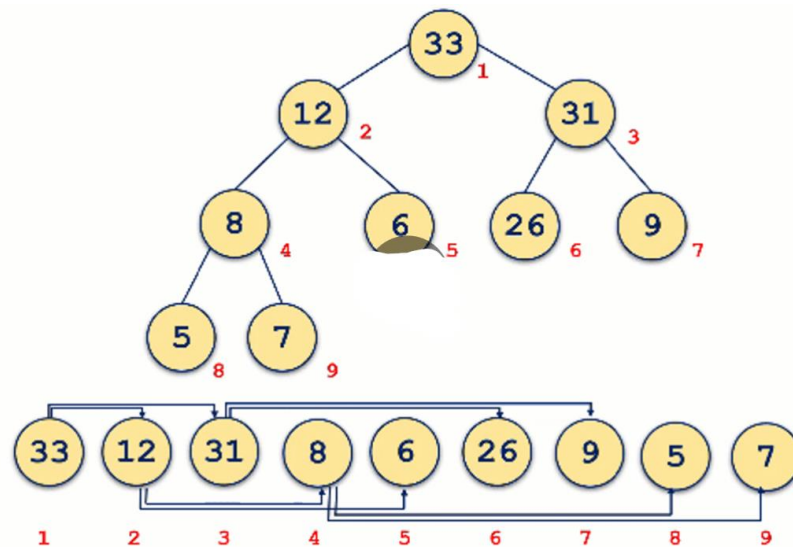


Figura 2.16. Legătura între heap și tabloul liniar cu n elemente

Unul dintre cei mai utili algoritmi pe structura de date heap este algoritmul de ordonare HeapSort. Algoritmul are la bază două operații standard, implementate pe Heap.

- Inserarea unui element în Heap. Operația se execută nemijlocit în tabloul care modelează heap-ul prin o serie de interschimbări a pozițiilor elementelor în concordanță cu proprietățile declarate ale heap-ului
- Excluderea unui element din Heap. Operația este însoțită de excluderea elementului cu valoare maximală din rădăcină și înlocuirea cu ultimul element din heap, apoi – restabilirea proprietății de Heap. Complexitatea și eficiența algoritmului sunt de același ordin ca a algoritmului MergeSort – $O(n \times \log(n))$

Arborii indexați binar (arborii Fenwick) - sunt structuri de date care pot actualiza eficient elementele și pot calcula sumele pe intervale de indici într-o listă de numere. Sunt cele mai populare structuri de date utilizate pentru modificarea dinamică a datelor.

Algoritmi probabilistici – algoritmi tip Monte-Carlo care permit calcularea valorilor numerice în baza unor experiențe probabilistice, prin verificarea unui număr suficient de mare de experiențe aleatorii și clasificarea rezultatelor acestora. Metodele pot fi utilizate în cele mai diverse contexte în științele naturii pentru calcularea ariei, volumului, a altor măsurători cantitative, care tind spre un rezultat veridic după un număr suficient de mare de repetări.

Problema- tip : *problema de calculare a suprafețelor poligoanelor Voronoy pe plan, fără a construi poligoanele propriu-zise* [166], [32], [148].

Metode euristice – se revine la metoda Greedy, pentru a analiza situațiile în care aceasta este eficientă dar nu permite obținerea celei mai bune soluții decât aleator, în funcție de caz – pentru problemele din clasa NP. Exemplu – metoda Greedy pentru colorarea grafului [167]. Sau,

o problemă echivalentă ca și importanță – metoda Greedy aplicată la rezolvarea problemei rucsacului.

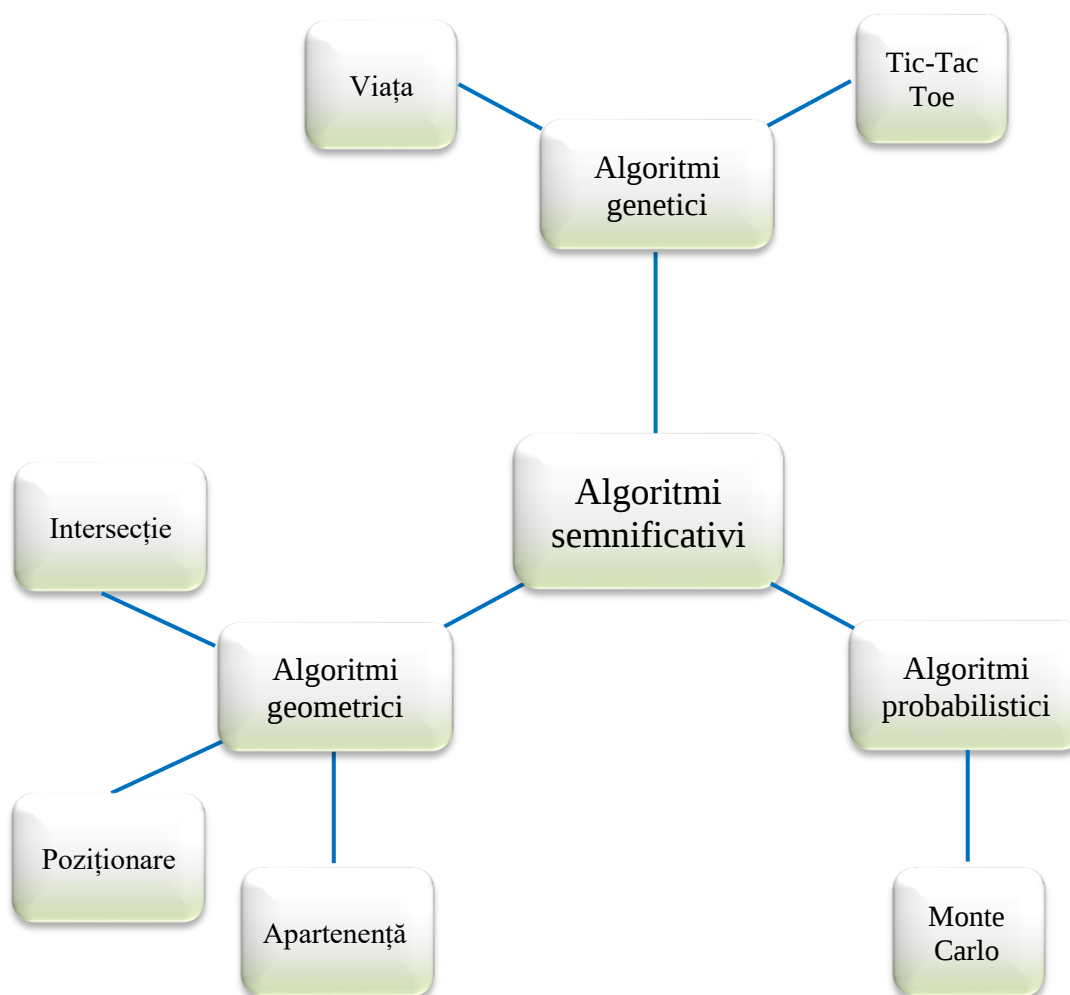


Figura 2.17. Algoritmi semnificativi

Probleme tip pentru această metodă sunt problemele în care se solicită achitarea unei sume la bancomat, folosind un număr minimal de bancnote (problema bancomatului infinit și problema bancomatului real – cu cantități restricționate ale bancnotelor), sau bine-cunoscuta problemă a hoțului de chibrituri: un hoț a pătruns într-un depozit și vrea să fure cât mai multe chibrituri. În depozit se află m containere. Containerul cu numărul i conține a_i cutii de chibrituri, iar în fiecare cutie se află b_i chibrituri. Toate cutiile din container au dimensiuni egale. În rucsacul hoțului încap exact n cutii. Determinați cantitatea maximală de chibrituri pe care o va putea pune în rucsac hoțul. Hoțul nu mută chibrituri dintr-o cutie în alta din lipsă de timp. El doar va lua nu mai mult de n cutii astfel încât cantitatea totală a chibriturilor din cutii să fie maximală.

Date de intrare: prima linie a datelor de intrare conține numerele n ($1 \leq n \leq 2 \cdot 10^8$) și m ($1 \leq m \leq 20$), separate prin spațiu. Linia $i + 1$ conține perechea de numere a_i și b_i ($1 \leq a_i \leq 10^8$, $1 \leq b_i \leq 10$). Toate numerele din datele de intrare sunt întregi.

Date de ieșire: afișați în output-ul standard un număr întreg – numărul maximal de chibrituri pe care le poate scoate din depozit hoțul. (2019. Problemă propusă la Olimpiada pentru Colegii, zona Centru-Sud).

Metoda Greedy poate să apară și ca o parte, etapă, în rezolvarea unei probleme mai complexe. Tocmai așa se întâmplă în exemplul deja utilizat în lucrare – problema Coroziunea Metalului (pag. 69).

O a doua categorie de metode euristice este cea a **λ -aproximărilor**. Se folosesc în special pentru problemele discrete, pentru a identifica grupuri de elemente care influențează cel mai puternic un sistem : [27] [143]. Se folosesc pentru probleme de căutare pe grafuri , pentru identificarea k – centrelor și k – medianelor în grafuri. Respectiv pot ajuta la calcularea k - razei grafului.

Problemă-tip: (Olimpiada zonală între colegii, zona Centru-Sud, 2012. Megașcoala) *Pe insula Tortuga sunt N localități (numerotate de la 1 la N). Guvernatorul insulei, fostul corsar Ion Vrabie, a hotărât să construiască două megașcoli moderne, la care vor merge copii din toate localitățile. Copiii din fiecare localitate urmează să meargă la cea mai apropiată dintre megașcoli. Localitățile în care vor fi construite megașcolile se vor selecta astfel, încât timpul necesar pentru a ajunge la megașcoală din cea mai îndepărtată localitate să fie cât mai mic posibil.*

Cerință: fiind date timpurile de deplasare între toate localitățile insulei, să se scrie un program care să determine indicii localităților, în care se vor construi megașcolile.

Ultima categorie de euristici, care se studiază în cadrul instruirii la etapele C1- C3 este categoria algoritmilor genetici. Algoritmii genetici sunt tehnici de căutare, identificare, optimizare a soluțiilor, pentru rezolvarea problemelor inspirate din procesul natural de evoluție și selecție. Algoritmii genetici au la bază ideea de a genera și îmbunătăți soluții într-un mod similar cu modul în care evoluează organismele în natură.

Principiul general de funcționare a algoritmilor genetici:

1. Se pornește de la o populație inițială de soluții (cromozomi sau indivizi).
2. Fiecare individ este evaluat după un criteriu de fitness (calitatea soluției).
3. Se selectează cei mai buni indivizi pentru a produce următoarea generație de soluții.
4. Soluțiile noi se generează prin operații precum încrucișarea și mutația.
5. Procesul se repetă pe mai multe generații, având ca scop îmbunătățirea continuă a soluțiilor, până în momentul când un număr semnificativ de generații a soluțiilor nu mai generează un progres sau optimizare.

Problema tip, cu care începe studierea acestei clase de algoritmi este problema jocului Viața [88, p. 458] (The Game of Life): *sensul jocului este ca, pornind de la o configurație inițială*

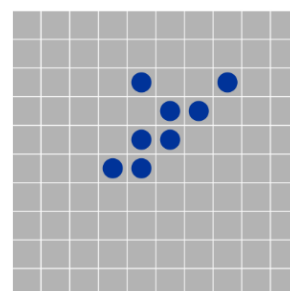
a unei populații imaginare să se modeleze transformările acestea în funcție de un set de „legi genetice”, care controlează nașterea, moartea sau supraviețuirea indivizilor. Aceste legi au fost formulate de Conway după îndelungate verificări și teste, astfel încât să satisfacă într-o măsură cât mai mare următoarelor trei condiții:

- Nu trebuie să existe nici o configurație inițială pentru care să existe o demonstrație simplă a posibilității de dezvoltare perpetuă a populației.
- Trebuie să existe configurații inițiale, care posedă apriori proprietatea de dezvoltare perpetuă a populației
- Trebuie să existe configurații inițiale simple, care, timp de un număr suficient de mare de mișcări se dezvoltă, se modifică și își sfârșesc procesul evolutiv în unul din următoarele trei moduri: dispar complet; se transformă într-o configurație stabilă și nu se mai modifică; sau capătă o perioadă fixă de transformare, revenind la o una și aceeași stare după un număr constant de mișcări.

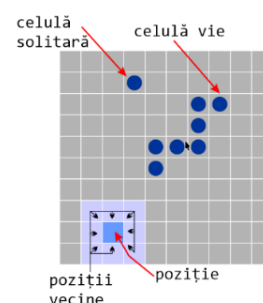
Altfel vorbind – comportamentul populației în timp trebuie să fie imprevizibil!

Așa dar „Viața” este un joc de modelare a proceselor de apariție, evoluție și moarte a coloniilor de organisme vii, sau, mai general – a proceselor care au loc în natura vie.

În imaginea alăturată avem o configurație inițială posibilă a jocului. Pozițiile cu discuri albastre reprezintă celulele (indivizii) vii ale populației. Tabla se consideră infinită, dar, de cele mai multe ori este suficientă o porțiune destul de mică pentru a urmări evoluția populației



Clarificări: **poziție** este numit un pătrățel elementar distinct de pe tabla de joc; două **poziții** se numesc **vecine**, dacă au o latură sau un vârf comun. Astfel, fiecare celulă are opt poziții vecine, respectiv numărul celulelor vecine poate fi mai mic sau egal cu opt; **celula** este **solitară** dacă nu are nici un vecin

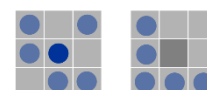


Regulile de joc: la etapa zero pe tablă se construiește o configurație inițială. Apoi urmează o serie de transformări ale celulelor, fiecare dintre care este realizată în baza configurației din etapa precedentă. Regulile de transformare sunt simple:

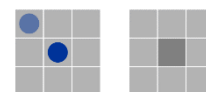
Dacă la etapa curentă celula este vie și are în calitate de vecini 2 sau 3 celule vii, la etapa următoare celula vă rămâne vie.



Dacă la etapa curentă celula este vie și are în calitate de vecini mai mult de 3 celule vii, la etapa următoare celula vă muri din cauza suprapopulării.



Dacă la etapa curentă celula este vie și are în calitate de vecini nu mai mult decât o celulă vie, la etapa următoare celula va muri din cauza singurătății.



Dacă la etapa curentă o poziție este liberă dar are în calitate de vecini exact trei celule vii, la etapa următoare în această poziție va apărea (se va naște) o celulă vie.



Sarcină: să se scrie un program, care pentru o configurație inițială dată a populației să determine starea ei peste n generații.

Algoritmii de **geometrie computațională** au dat naștere la o clasă nouă de probleme competiționale, care au diversificat mult structura probelor competițiilor de programare. Algoritmii geometrici se referă de obicei la probleme și situații în care se cere de calculat coordonatele unui punct(unor puncte) sau de verificat poziția reciprocă a două sisteme de puncte în plan.

Algoritmii fundamentali sunt foarte simpli: *determinarea poziției punctului față de un vector, calcularea distanței între două puncte și determinarea intersecției a două linii*. Celelalte probleme pot fi reduse la o serie de iterații a problemelor menționate, sau rezolvate prin metode probabilistice.

Cei mai cunoscuți algoritmi din spațiul competițional sunt: algoritmii de triangularizare (cu extensii în toate domeniile ce se ocupă de modelare), determinarea apartenenței la un domeniu(convex sau simplu), calcularea înfășurătorii convexe, algoritmii asociați poligonului Voronoy.

Probleme-tip: 1. Evadare [140] *Un grup de pinguini a decis să evadeze din grădina zoologică. În acest scop ei au săpat un tunel liniar. Din nefericire, zidul grădinii zoologice formează un poligon simplu cu N ($3 \leq N \leq 10000$) laturi, astfel încât, ieșind din tunel, pinguinii nu pot afirma cu certitudine dacă se află în interiorul grădinii zoologice sau în afara ei. Să se scrie un program care, după coordonatele punctelor extreme ale tunelului și coordonatele vârfurilor poligonului ce stabilesc perimetrul grădinii zoologice, va determina dacă evadarea s-a soldat cu succes sau nu. (Soluție – algoritmul de apartenență la domeniu)*

2. Turnuri [166] *Bitlanda este o țară care se extinde permanent, liniar. Frontiera țării reprezintă o linie frântă închisă, fără autointersecții. Pentru a apăra frontierele sale, după fiecare extindere, în noile puncte extreme ale frontierei se construiesc turnuri de veghe. Există N turnuri de veghe, date prin coordonatele lor (x_i, y_i) – numere întregi. Regele Bitlandei, Bytezar, a decis să trimită la vatră garnizoanele turnurilor care nu se mai află la hotarele țării. Să se scrie un program care va determina câte turnuri vor fi lipsite de garnizoane. (Soluție – algoritmul înfășurătorii convexe)*

3. Carcasa [140] *Fie o carcasă având forma unui poligon simplu. Pentru a fi poziționată vertical pe o suprafață plană, carcasa trebuie să aibă centrul de masă situat strict între 2 puncte de contact cu suprafața. Centrul de masă este întotdeauna un punct interior al carcasei și nu coincide cu nici un vârf al ei. Să se scrie un program care va determina numărul pozițiilor în care poate fi stabilizată vertical carcasa.* (Soluție – algoritmul înfășurătorii convexe + principii fizice pentru determinarea centrului de masă)

Metodologia. Rezolvarea unor probleme complexe presupune un efort intelectual individual semnificativ, care urmează să fie antrenat în timp. Activitățile de bază vor fi activități individuale, sau de interacțiune cu sistemul automat de evaluare. Totuși, componenta inițială a instruirii va fi una tradițională, de transmitere a cunoștințelor teoretice și experienței practice de rezolvare a categoriilor specificate de probleme, ceea ce presupune și utilizarea unor metode didactice tradiționale, adaptate la specificul disciplinei: mentoratul personalizat, învățarea colaborativă, studii de caz în baza problemelor de concurs, rezolvarea de probleme – individuală și pe echipe. La final – interacțiunea inversă – autorul (autorii) unei soluții validate descriu etapele de realizare a sarcinii, ideile de optimizare folosite și punctele de cădere pe parcurs.

Selecția problemelor. Varietatea domeniilor din care provin problemele deschise (care nu au o soluție optimă, tangibilă în timp real sau aproape de acesta) garantează un număr suficient de mare de task-uri pentru exersare în condiții domestice. Acestea vor fi suficient de complicate pentru a asigura o soluție optimă doar prin combinarea reușită a modelului matematic corect, structurilor de date eficiente, intuiției și inteligenței.

Concursuri. La nivelul avansat de instruire concursurile de antrenament se formează în baza colecțiilor de probleme tematice de pe platformele de evaluare. În calitate de platformă primară da antrenament se recomandă CodeForces, secundar – InfoArena.

Poate fi valorificată și experiența concursurilor regionale, organizate în România și Bulgaria. În perioada de toamnă – concursul cu două zile de competiție - Romanian Masters in Informatics, primăvara – International Advanced Tournament in Informatics (IATI), Shumen, Bulgaria – la fel, două zile de competiție și „Urmașii lui Moisi”, Romania – o singură etapă.

Structura concursurilor de antrenament locale: 1 problemă complexă pentru 2 ore astronomice sau 2 probleme pentru 3 ore. Problemele de concurs vor avea o formulare care să permită abordări multiple în rezolvare, dar și formularea subproblemelor de complexitate eşalonată – fie după dimensiunea setului de date de intrare, fie după structura acestuia, fie după raportul valoric dintre componentele de date.

2.3. Specificul aplicării modelului CRYPTEX pentru instruirea de performanță la Informatică

Utilizarea mecanică a consecutivității de studiere a conținuturilor teoretice nu ar diferenția prea mult rezultatele utilizării modelului, în comparație cu un syllabus de performanță, cum ar fi, de exemplu [168]. Deosebirea principală este determinată de doi factori practici: colecțiile de probleme, exerciții și resurse auxiliare pe domeniul de bază al fabulei pentru fiecare problemă, și concursurile interne și externe, în special cele neoficiale, în care sunt antrenați elevii.

Colecții de probleme

Acest element de formare a performanței presupune accesul la setul integral de resurse, care formează o problemă informatică: enunț, exemple, soluție, baterii de teste pentru evaluare. Resursele tipărite sunt, de obicei, incomplete: chiar dacă conțin enunțul, soluția și exemplele necesare pentru înțelegerea problemei, nu pun la dispoziție nici bateriile de teste, nici sistemul de evaluare automată, or, acesta din urmă este o necesitate, atât timp cât problemele admit soluții multiple [169, pp. 73-85]. În ultimii ani, odată cu apariția platformelor de pregătire cu evaluare automată tip InfoArena [137], CodeForces [136], AtCoder [170], păstrarea copiilor locale ale acestor seturi de resurse nu mai este o condiție obligatorie. Totuși, sunt cunoscute situații când platformele web de evaluare încetează să mai funcționeze după o perioadă, ceea ce poate duce la pierderi de date, resurse valoroase, exemplu – prima platformă pentru pregătire de performanță românească - punct Campion [171]. Prin urmare, pentru fiecare problemă atribuită în calitate de exercițiu practic unei resurse teoretice se va păstra cel puțin arhiva locală.

Colecțiile de probleme, exerciții și resurse formează cea de a treia dimensiune a fiecărui compartiment teoretic, astfel, etapele circulare de acumulare a cunoștințelor teoretice se transformă în „discuri” cu o importantă încărcătură practică. Altfel vorbind, are lor proiectarea conținuturilor și resurselor teoretice pe gama de activități practice, integrarea lor și utilizarea ca un tot întreg, în perioada potrivită de instruire [172] [41].

La nivel de model, transformarea este una radicală, Figura 2.18.

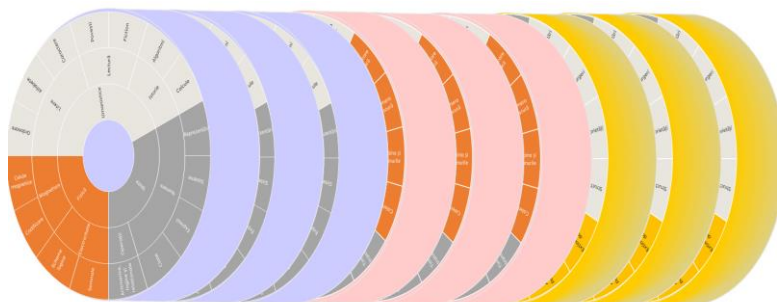


Figura 2.18. Modelul 3D. Conținuturile teoretice, Resurse pentru activități practice, colecții de probleme

Resurse de instruire interactive

Conținuturile studiate, în special cele de natură complexă, matematică sau fizică, pot fi explicate / înțelese mult mai eficient, dacă procesele explicate sunt modelate vizual. În special această afirmație se referă la componentele de natură fizică, algoritmi de sortare și la structurile de date.

Elaborarea unor resurse interactive, care să demonstreze un proces informatic nu este deloc simplă și cere un efort suplimentar considerabil din partea mentorului. În calitate de alternativă pot fi folosite resursele vizuale în distribuție liberă. Un exemplu de resursă elaborată pentru comunitate este simulatorul de circuite electrice [173].

În context informatic, sunt considerate resurse interactive orice sisteme informaționale, simulatoare, medii virtuale sau de altă natură, elementele cărora reacționează activ, în moduri distincte și prestabilite la interacțiunea cu factorul uman, oferind acestuia în urma interacțiunii posibilitatea să testeze cunoștințele, abilitățile sau competențele specifice în timp real [54] [55].

Un instrument interactiv de învățare, dezvoltat din aplicațiile de testare a soluțiilor informatice este, indiscutabil, orice platformă informatică, care oferă utilizatorilor săi posibilitatea de a se identifica și păstrează istoricul activității fiecăruia dintre ei. Prin istoric al activității se înțelege totalitatea acțiunilor realizate de utilizator pe platformă. În contextul informaticii competitive și a platformelor virtuale pentru concursurile de programare istoricul activității fiecărui elev pe o platformă distinctă permite urmărirea neinvazivă a procesului de formare, punctele slabe, momentele în care este necesară intervenția mentorului. Același mediu al platformelor de antrenament informatic poate fi folosit pentru organizare celor mai diferite modele de comunicare: cursant $\leftrightarrow$ comunitate informatică virtuală, cursant $\leftrightarrow$ mentor, cursant $\leftrightarrow$ cursant, cursant $\leftrightarrow$ sistem digital.

Mai mult, platformele informatice de instruire pot fi considerate meta resurse, deoarece dispun de mai multe instrumente de învățare integrate: colecții de probleme, instrumente de evaluare, librării digitale specializate, alte resurse opționale. Un exemplu recent de arhitectură și conținut selectat cu grijă este platforma punct Campion, una dintre puținele care oferă posibilitatea de a studia interactiv principiile de funcționare a diferitor categorii de algoritmi, inclusiv algoritmi pe grafuri, care se referă la construcția arborilor parțiali de cost minim, calcularea distanțelor minimale în graf, construcția ciclurilor Euler și Hamilton, dar și algoritmi de geometrie computațională pentru determinarea apartenenței la un domeniu oarecare, construirea înfășurătorii convexe, identificarea celei mai apropiate perechi de puncte sau dimpotrivă a celei mai distanțate perechi de puncte în plan.

O categorie aparte o formează platformele vizuale cu resurse interactive de învățare pentru programatorii începători. Specificul acestora este posibilitatea de construcție a programului din blocuri predefinite, urmărirea execuției instrucțiunilor pas cu pas, observarea în timp real al erorilor logice și constructive [174].

Prin urmare, combinarea activităților practice tradiționale, în sala de clasă sau în laboratorul informatic cu utilizarea, la primele etape de instruire a platformelor vizuale de programare, iar apoi a platformelor virtuale de instruire aduce o valoare adăugată întregului proces de formare a elevului, capabil de performanță informatică.

Concursuri

Forma actuală a modelului, chiar dacă este suficient de consistentă, rămâne instabilă din punct de vedere a unui sistem integrat – etapele de formare sunt separate unele de altele, oferind multiple posibilități de destabilizare, în sens de pierdere a motivației – fiecare etapă poartă o încărcătură cognitivă considerabilă, presupune eforturi intelectuale continue și majore, un mod de gândire divergent.

Stabilizarea modelului de instruire, asigurarea motivației, poate fi obținută în diverse moduri. Logic este să fie folosite chiar instrumentele care stau la dispoziția oricui, în distribuție liberă, gratuită, sau care implică cheltuieli nesemnificative. Axul de stabilizare a modelului este format din participări ale elevilor la concursuri, fie acestea on-line, fie cu prezență, locale, regionale sau chiar internaționale.

Există diverse tipuri de concursuri informatice de programare care pot forma tija de stabilizare a modelului. Mai multe dintre ele se desfășoară on-line sub egida unor organizații neguvernamentale sau a unor echipe de entuziaști, de multe ori – foști olimpici. A fost menționată anterior platforma CodeForces – aceasta organizează cu intervale de două – trei săptămâni runde de concurs, la care doritorii din întreaga lume pot participa on-line. Mai mult, rundele sunt plfonate pe nivele de complexitate a problemelor, formând patru „div-uri⁶” distincte. Participanții pot să aleagă cel mai confortabil nivel de complexitate și să participe în concursurile de categorie respectivă. Participarea continuă la rundele de concurs este stimulată de atribuirea unui rating cumulativ.

O a doua categorie de concursuri recomandate o formează concursurile organizate mixt sau on-site, cu fond de premiere. Acestea sunt organizate de instituții de învățământ superior, împreună cu companii din domeniu, având o misiune dublă: pe de o parte promovarea elevilor cu performanțe în domeniul programării, pe de alta – promovarea propriei imagini. Concursuri-

⁶ Fiecare div stabilește un nivel de complexitate al problemelor propuse în competiție.

tip de acest gen sunt „The Game of Code” (Republica Moldova), IATI (Bulgaria) [175], concursul de programare al Universității Babeș-Bolyai, Cluj.

Cea de a treia categorie de concursuri (neoficiale pentru participanții din Republica Moldova) sunt concursurile regionale din România și alte țări, de exemplu, „Urmașii lui Moisi” - România, concursurile internaționale regionale, la care Moldova nu este cooptată în calitate de membru – Olimpiada Țărilor Europei Centrale, dar la care poate participa sporadic, în funcție de prezența unei invitații din partea țării organizatoare. Aceste concursuri sunt deosebit de importante, deoarece contribuie și la formarea unor legături directe cu participanții din alte țări, dezvoltând astfel o comunitate internațională a tinerilor pasionați de informatică. Integrarea elevilor în această comunitate este extrem de importantă, deschizând pentru ei cea mai eficientă metodă de învățare pentru performanță „peer-to-peer learning” [10], [176], [177].

Un moment important, care urmează a fi menționat, este că participarea la concursuri nu trebuie limitată doar la concursurile informatice. Dimpotrivă, se vor încuraja participările elevilor la concursuri de fizică, matematică și alte discipline, organizate în Republica Moldova, dar și în afara ei. Cel mai relevant exemplu, în acest context, este participarea la concursul de fizică „În memoriam Mihai Marinciuc”, organizat de Universitatea Tehnică a Moldovei [178], *Concursul de Matematică al liceului Orizont*.

Toate aceste categorii de concursuri au aceeași proprietate specifică – succesul la o asemenea competiție este perceput de elevi mai puternic decât în cazul competițiilor oficiale – doar posibilitatea de a participa este deja percepută ca fiind un succes „garantat”, în timp ce percepția nereușitei este atenuată de procesul de participare la o competiție dedicată, integrare în comunitatea informatică, socializare și impresii noi. În consecință, concursurile periodice, neoficiale, de la nivel local până la cel internațional, doar participarea la care necesită un efort suplimentar din partea elevilor, mentorului, a instituției de învățământ, formează axul și elementul central al modelului instruirii de performanță, prezentat în lucrare (Figura 2.1, C, D).

Produsul final al modelului elaborat este *elevul performer* – noțiune introdusă de Jozeph Renzulli în articolul „*The Three-Ring Conception of Giftedness: A Developmental Model For Promoting Creative Productivity*” [179]. În publicațiile din Republica Moldova subiectul a fost dezvoltat de cercetătoarea Tatiana Callo, care a descris și profilul general al elevului performer [180]. Profilul particular al elevului performer la disciplina Matematica a fost conturat în articolele cercetătorilor Ilie Lupu, Larisa Sali, Marcel Teleucă [181], [182]. Rafinarea acestui profil pentru disciplina Informatică, a rămas însă, până acum, o problemă deschisă.

2.4. Profilul elevului performer la Informatică

Evoluția rapidă a societății informaționale și a tehnologiilor de programare ridică competențele în domeniul informaticii la nivel de competențe esențiale pentru dezvoltarea personală și academică a elevilor. Identificarea și înțelegerea caracteristicilor elevului performer în această disciplină reprezintă un element fundamental pentru optimizarea procesului educațional și pentru promovarea excelenței în domeniu. Cunoașterea profilului elevului performer permite personalizarea și adaptarea metodelor didactice, contribuie la dezvoltarea unor strategii eficiente de motivare și de formare a celor mai talentați tineri informaticieni.

Concursurile de programare ocupă un loc central în identificarea, stimularea și dezvoltarea potențialului elevilor talentați. Participarea la astfel de competiții le oferă tinerilor oportunitatea de a-și pune în valoare abilitățile, de a-și testa limitele și de a se confrunta cu probleme reale, dezvoltând astfel gândirea critică, creativitatea și perseverența. Concursurile contribuie, totodată, la crearea unui mediu competitiv și motivant, care stimulează excelența academică și profesională în domeniul informaticii.

În acest context identificarea profilului elevului performer la disciplina Informatică are și importanță pedagogică majoră, deoarece permite fundamentarea teoretică și practică a intervențiilor educaționale. Astfel, profilul tip devine un instrument valoros în selectarea și pregătirea elevilor capabili să răspundă provocărilor Informaticii competitive moderne. În plus, înțelegerea componentelor profilului elevului performer și utilizarea lor creativă ajută la identificarea factorilor care stimulează excelența în domeniul informaticii, contribuind la îmbunătățirea constantă a rezultatelor acestuia în cadrul competițiilor informatice.

Structura profilului general al elevului s-a format în timp, pornind de la cercetările lui Benjamin Bloom [95], dezvoltată de Howard Gardner [183] și Albert Bandura [184] și adaptată la realitățile societății informaționale de același Howard Gardner [185], Dale Schunk [186] și mulți alți cercetători. Astfel, au fost identificate în timp următoarele componente ale profilului elevului:

- **Componenta cognitivă**, care se referă la ansamblul proceselor mentale implicate în învățare: percepția, atenția, memoria, gândirea, rezolvarea de probleme și limbajul. De aceeași componentă țin și abilitățile analitice: capacitatea de analiză, sinteză și judecată, nivelul de dezvoltare intelectuală a elevului [187].
- **Componenta motivațională**, care se referă la motivația intrinsecă - dorința de a învăța pentru plăcerea cunoașterii - și motivația extrinsecă (recompense externe, note, opinii sociale). Nivelul de motivație influențează angajamentul, insistența și efortul depus în procesul de învățare, are un impact direct și major asupra performanței în domeniile care sunt acoperite motivațional [188], [184].

- **Componenta afectivă.** Aceasta se referă la emoțiile, atitudinile și valorile care influențează comportamentul și motivația elevului. Componenta include: încrederea în sine, autorespectul, sentimentele de satisfacție pentru succes, depășirea anxietății și stresului, sentimentul de autoapreciere. Aceste elemente determină modul în care elevii percep provocările și depășesc dificultățile [184], [189], [190].
- **Componenta comportamentală.** Aceasta cuprinde stilurile de învățare, rutine și deprinderi, modelul de autoorganizare, perseverența și responsabilitatea atât în activitățile educaționale, cât și în cele cotidiene. Se referă la comportamentele observabile tip: gestionarea timpului, autoorganizarea, capacitatea de a acționa autonom, și responsabilitatea pentru acțiunile sale în mediul educațional, dar și în afara lui [186], [191].
- **Componenta socială.** Se referă la abilitățile de interacțiune cu colegii și profesorii în context educațional, cu personaje arbitrare în situații cotidiene, comunicare generală și în grupuri de interese, colaborare tematică și generalistă, inclusiv la distanță. Include capacitatea de a lucra în echipă și spiritul de echipă, elemente de empatie, capacitatea de a negocia, acceptarea normelor sociale. Aceste aspecte sunt fundamentale pentru dezvoltarea unui comportament adaptiv și flexibil în mediul educațional și în afara lui [41].

Proiectarea modelului de instruire pentru performanță în Informatică dezvoltat de autor pe componentele profilului clasic conduce la următoarele grupuri de elemente ale profilului specializat pentru elevul performer la disciplina Informatica:

1. **Componenta cognitivă** a elevului performer la disciplina Informatică se manifestă prin:
 - a. **Abilități tehnice fundamentale:** care se referă la cunoașterea profundă a arhitecturii sistemelor informatice, sistemelor de operare, instalării și customizării aplicațiilor, ajustării parametrilor de funcționare a echipamentelor periferice. Un alt grup de abilități tehnice fundamentale este cel al abilităților de utilizare a limbajelor de programare de nivel înalt, în special al limbajelor de programare procedurale, tip C, C++, C#. Conform cercetărilor în domeniul educației și teoriei informației, o bază solidă în aceste elemente este esențială pentru dezvoltarea gândirii algoritmice și pentru capacitatea de a construi soluții eficiente. Mai multe studii, ca, de exemplu, studiul grupului de autori de la Colegiul de Informatică a Universității Maryland „Formarea gândirii Informatică la lecțiile de Matematică și Științe” [192, pp. 127 - 147] demonstrează corelarea puternică între achiziția de cunoștințe tehnice, performanțele academice, și abilitatea de a aplica aceste cunoștințe în situații practice.
 - b. **Abilități de rezolvare a problemelor complexe:** elevul performer are capacitatea de a transforma problema cotidiană sau de cercetare într-un model matematic, apoi informatic,

aplicând conceptele fundamentale ale informaticii: modelarea, algoritmi, structurile de date, programarea și matematica discretă. Este capabil să estimeze complexitatea problemelor, să identifice soluții optimale și să demonstreze corectitudinea acestora prin metode matematice (analitice) sau computaționale. Are o capacitate sporită de concentrare și mobilizare în timp pentru rezolvarea unor probleme complexe, fără soluții evidente. Altfel spus, are un aparat dezvoltat al gândirii critice și a celei sistemice. În general, noțiunea de „gândire algoritmică” reprezintă o competență fundamentală pentru învățarea și rezolvarea problemelor în societatea digitală. Setul de abilități integrat în aceasta permite elevilor să construiască modele și algoritmi eficienți, fiind esențial pentru succesul în domeniu [193], [194].

- c. **Cunoștințe fundamentale în științe reale, lingvistică și științe umanistice.** Prezența acestora rezultă din conținuturile studiate în cadrul implementării modelului (matematică, fizică - în special fizică computațională, elemente computaționale ale științelor biologice, chimiei). Este esențială cunoașterea limbii engleze, dar și a limbii române pentru înțelegerea corectă a textelor informatice și enunțurilor problemelor în condiții de concurs, contra cronometru, sau în condiții cotidiene.
 - d. **Creativitate:** Elementul creativ se manifestă prin capacitatea de a identifica soluții inovatoare pentru probleme complexe, inclusiv în baza unor algoritmi euristici, dezvoltați intuitiv, efectuând unele analize parțiale a datelor și prin capacitatea de evaluare calitativă a soluțiilor particulare. Profilul elevului creativ la disciplina Informatică va conține neapărat elemente de gândire / inteligență divergentă care presupune evaluarea și analiza logică a informației, fiind fundamentală pentru dezvoltarea soluțiilor eficiente și pentru adaptarea rapidă la noi tehnologii. În contextul învățării informatice, aceste abilități permit elevilor să dezvolte idei inovatoare și să se adapteze instant la situații noi [195], [196].
2. **Componenta motivațională** a profilului elevului instruit prin intermediul modelului CRYPTEX se manifestă atât intrinsec, cât și extrinsec:
- a. Motivația intrinsecă a elevilor este un factor critic în atingerea performanței. Conform teoriei auto-determinării a lui Deci și Ryan [188], elevii motivați intrinsec sunt mai predispuși să investească timp și efort în progresul lor, manifestând persistența și angajamentul necesar pentru exersare și învățare profundă. În contextul disciplinei de studiu Informatică, amplificarea interesului și motivației este determinată de posibilitatea dezvoltării propriilor proiecte informatice în baza utilizării creative a instrumentelor software, aplicate în procesul de studiu a Informaticii competitive. Acestea din urmă reprezintă în majoritatea cazurilor aplicații utile fie pentru comunitate (site-uri web, produse de design grafic, agenți de

Inteligență Artificială), fie pentru mediul educațional: resurse de învățare interactive, sisteme de automatizare a proceselor instituționale, baze de date locale sau de rețea pentru evidența unor bunuri, activități, evenimente sau fenomene. Recunoașterea valorii acestor produse prin utilizarea lor generează un efect intrinsec puternic. La fel de importantă este și recunoașterea publică a valorii rezultatelor obținute de elevi în cadrul competițiilor naționale și internaționale. La amplificarea efectului intrinsec pot contribui administrațiile de toate nivelurile ale sistemului educațional – de la instituția de învățământ până la Ministerul Educației și Cercetării, Guvern și Președinție. În acest context este important de menționat și efortul permanent al structurilor guvernamentale pentru realizarea activităților de promovare a performanței în educație prin Gala Olimpicilor, organizată anual de Agenția pentru Evaluare și Curriculum și Ministerul Educației și Cercetării, întâlnirile elevilor olimpici cu Președintele Țării, alte activități festive, care aduc în prim plan rezultatele performanței școlare.

- b. Motivația extrinsecă este organic conectată la cea intrinsecă, deoarece recunoașterea externă a meritocrației elevului stimulează imediat motivația internă, dorința elevului de a-și extinde limitele cunoașterii, de a-și dezvolta capacitățile și performanța. De cele mai multe ori acțiunile de motivare intrinsecă și extrinsecă alternează, creând un mediu motivațional consistent: succesul nematerial, realizat prin obținerea unei medalii sau mențiuni de onoare la o competiție internațională este urmat de o activitate publică de premiere la nivel național [197]. Componenta de motivare extrinsecă în cazul competițiilor neoficiale poate fi extrem de variată: de la contribuția instituției de învățământ la organizarea plecării la o competiție în afara țării (Romanian Masters of Informatics, Urmașii lui Moisil, IATI) până la cadouri și premii din partea sponsorilor competițiilor, dar indiferent de modul de realizare, aceasta are rolul de a menține și dezvolta pasiunea elevului pentru cunoaștere.
3. **Componenta afectivă** urmează specificul profilurilor elevilor performeri la disciplinele reale: fiind familiarizați cu tehnici de demonstrație a adevărului științific, în prim plan apar:
 - a. **Valorile** care influențează comportamentul și motivația elevului. Capacitatea de a demonstra matematic sau în alt mod științific corectitudinea poziției sale (științifice sau de altă natură) generează o încredere în sine argumentată, și stare de autorespect;
 - b. **Emoțiile** sunt mai puțin vizibile. Rolul lor este diminuat de capacitatea de a gândi structurat și de caracterul analitic al gândirii elevului performer în Informatică. Totuși, sentimentul de satisfacție pentru succes, sau, dimpotrivă, insatisfacție din cauza insuccesului poate sparge barajele argumentelor logice, provocând stări de euforie sau depresie – ambele fiind la fel periculoase, în special în cazul apariției într-o etapă intermediară a competițiilor.

- c. **Atitudinea** elevului performer la disciplina Informatică față de educație, știință, societate, activități, evenimente și fenomene se formează prin trecerea acestora prin prisma valorilor sale, a gândirii specifice și transpunerii frecvente a situațiilor în care urmează să ia o atitudine, în modele discrete binare.
4. **Componenta comportamentală** este de asemenea în mare parte similară componentei comportamentale a elevilor performeri la oricare din științele exacte, dar totuși are particularități generate de specificul instrumentelor de învățare, utilizate pentru performanța în Informatică:
- a. **Stilurile de învățare**, chiar dacă sunt individuale pentru fiecare dintre elevi, în cazul informaticii presupun capacitatea de a îmbina noțiunile din diverse domenii în modele integrate, care apoi să fie utilizate fie pentru rezolvarea unor probleme, fie pentru obținerea unor cunoștințe teoretice noi. Stilul de învățare tipic al elevului performer la disciplina Informatică este în primul rând activ – elevul este în căutarea informațiilor noi, pe care le transformă în cunoștințe, combinând abordarea secvențială și globală.
 - b. **Deprinderile**, legate de modelul de învățare, țin de organizarea și priorizarea activităților de învățare, fiind conectate la **modelul de autoorganizare**, care se formează la elev în contextul necesității de alocare a unor intervale semnificative de timp pentru studii extra curriculare. Tot aici apare încă o caracteristică specifică performanței – capacitatea de gestionare a timpului, iar generalizând cele enumerate – capacitatea de a fi autonom – posibil cea mai importantă caracteristică, ținând cont de posibilitatea instruirii informatice prin intermediul platformelor automate.
 - c. **Perseverența și responsabilitatea** sunt trăsături specifice ale elevului performer, care are de studiat materii și de rezolvat probleme care depășesc mult nivelul de complexitate a problemelor școlare. Situațiile de nereușită în studierea unui subiect sau rezolvarea unei probleme pot interveni cu atât mai frecvent, cu cât mai complexă este tema teoretică sau activitatea practică. În acest sens, tot mai importantă devine capacitatea de a relua procesul de învățare, de a identifica punctele critice în învățare, de a finaliza misiunile care rezultă atât din activitățile educaționale, cât și din cele cotidiene.
5. **Componenta socială** presupune analiza modelelor de:
- a. **Interacțiune directă** a elevului performer cu colegii și profesorii în context educațional, care, în general, este determinată de prezența sau lipsa în instituția de învățământ a unei comunități informatice: cu cât această comunitate este mai extinsă, cu atât mai predispuși spre comunicare sunt membrii ei. Lipsa unei comunități informatice în instituție poate genera frustrări, sentimente de subapreciere, pierderea interesului față de disciplina de studii sau față de performanță în general.

- b. **Interacțiune indirectă**, pasivă, prin intermediul resurselor asincrone – blogurilor, forumurilor, chat-urilor tematice, Elevii performeri la Informatică formează în timp abilități puternice de comunicare asincronă, dezvoltă capacitatea de a înțelege rapid modelele de exprimare ale profesioniștilor în domeniu, cunosc bine câteva limbi străine, neapărat – engleza. Posedă capacitatea de a identifica modelele de comunicare cu sisteme informatice complexe, interactive, dar și cu sisteme distribuite tip om – sistem informatic – om, fapt important în procesul de adaptare la interfețele sistemelor de evaluare în cazul concursurilor informatice. Mai mult, dezvoltă un model divergent de comunicare care permite distribuirea atenției și resurselor intelectuale în timpul concursului în paralel pentru rezolvarea problemelor, comunicarea cu sistemul de evaluare informatic și comunicarea cu juriul uman prin intermediul sistemului informatic.
- c. **Formare a grupurilor de interes**, care presupun o colaborare tematică sau generalistă, directă sau la distanță. Formează capacitatea de a lucra în echipă, spiritul de echipă, elemente de empatie, abilități de negociere, capacitatea de a accepta normele sociale. Integrarea în asemenea grupuri este o sarcină destul de complexă, pe care elevul urmează să o soluționeze împreună cu mentorul, cel puțin la etapa inițială de integrare. În calitate de soluție vin participările live la concursurile de programare, unde elevii comunică între ei, stabilesc legături, care apoi se cristalizează în activități comune de învățare, preponderent la distanță. Sarcina mentorului în acest caz este organizarea participării elevilor la asemenea concursuri. Un alt model de formare a grupurilor de interes sunt chat-urile asociate platformelor automate de învățare și evaluare informatică. Acestea sunt dotate cu instrumente colaborative, care facilitează discuțiile informatice în grup și comunicarea între membrii acestuia. În timp elevul selectează acele grupuri tematice, care corespund cel mai bine profilului său cognitiv, de comunicare, lingvistic și emoțional.

În concluzie, profilul elevului performer la disciplina Informatică este un model complex de abilități, valori și atitudini care pot fi dezvoltate doar prin o abordare sistemică, de durată, bine planificată și organizată, adaptată la specificul fiecărui elev individual, realizată de unul sau mai mulți mentori, cu inserții ale învățării informale în cadrul grupurilor de interes și a comunităților informatice (Figura 2.19).

În contextul specificului instruirii de performanță la Informatică, a rolului și abilităților profesorului (mentorului) care gestionează procesul de instruire propriu zis, la momentul actual poate fi creat și profilul acestuia, care va fi în mare parte identic cu cel al elevului performer. Ținând cont de faptul că perioada de expunere a mentorului la stresuri și provocări este net superioară perioadei de expunere a elevului, pot fi deduse mod direct și riscurile profesionale, aferente

activității de mentorat pentru Informatica Competitivă: epuizare și stres, ardere profesională, conflicte sau dificultăți de comunicare care pot degenera în frustrări sau tensiuni în comunicarea cu colegii, familia, impact negativ asupra stării de sănătate fizică și mentală, riscurile de responsabilitate legală, în special în cazul însoțirii echipelor de minori la olimpiade și concursuri care se desfășoară peste hotarele țării.

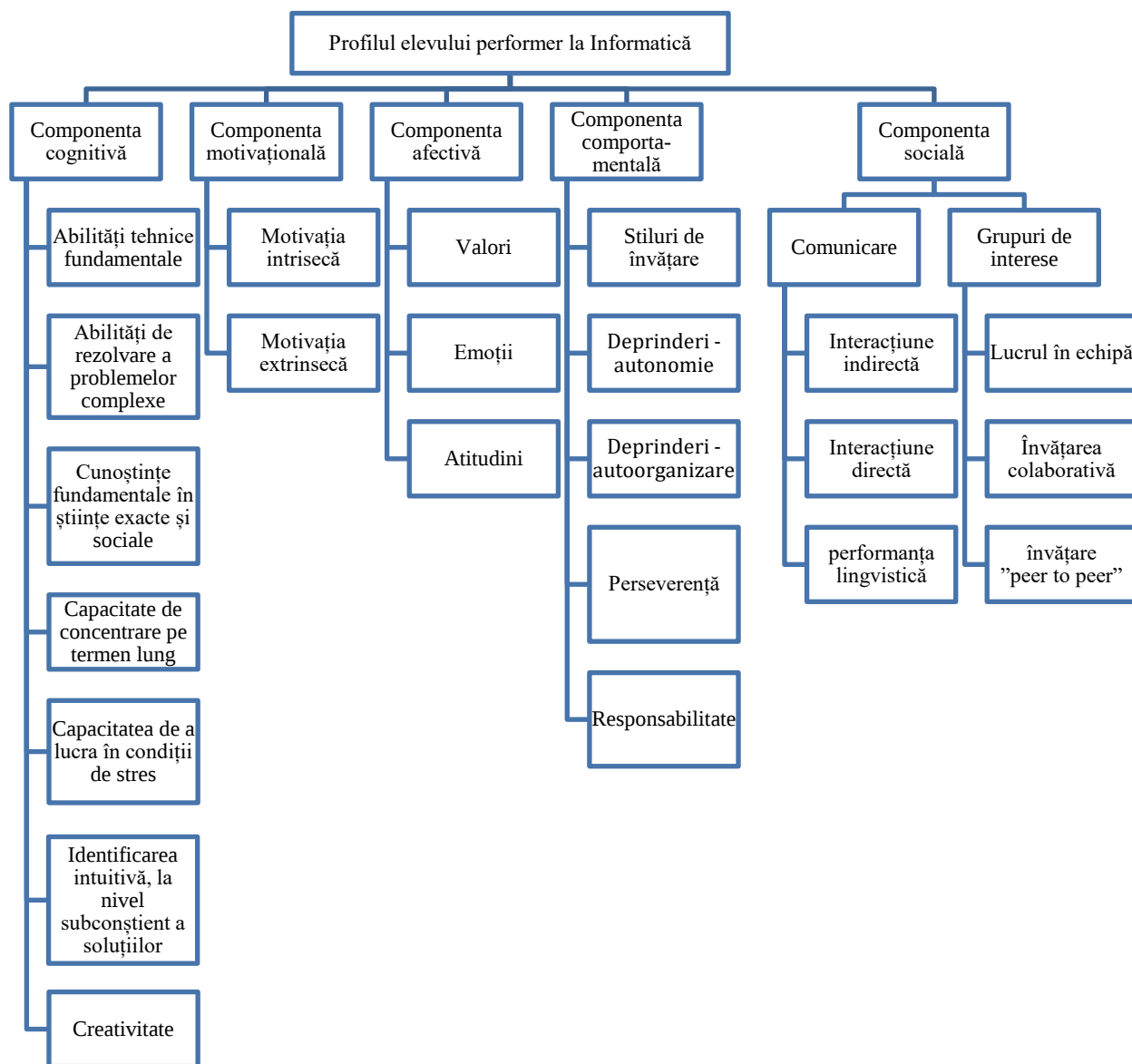


Figura 2.19. Profilul elevului performer la disciplina Informatica

2.5. Concluzii Capitolul 2

Modelul elaborat este unul complex. Implementarea lui cere un efort considerabil nu doar din partea celor instruiți, dar și din partea mentorului (sau mentorilor), care trebuie să se simtă confortabil în multitudinea de subiecte abordate, cu o tematică, variația căreia depășește standardele de cunoaștere pentru mai multe discipline școlare concomitent. Dar tocmai acest moment face ca modelul să fie adaptabil, flexibil, individualizat și să asigure un traseu optimal spre succes, succesul însemnând aici nu doar rezultatele la competițiile școlare, dat și performanța ulterioară la etapa studiilor universitare și a carierei profesionale ulterioare. În concluzie:

1. Modelul de instruire propus este fundamentat în zona de conținuturi de un Syllabus internațional, în zona de activități - de resurse educaționale moderne, inteligente și disponibile practic în orice împrejurări, de activități practice permanente, în zona de motivație - de posibilitatea reală de se simți și de a deveni parte a unei comunități informatice elitare.
2. Divizarea modelului pe nivele și etape permite fiecărui instruit să atingă limita individuală superioară de succes în informatica de performanță, sau să aleagă un traseu paralel în zona științelor reale, în special în domeniul matematicii competitive sau a fizicii de concurs. În același timp, divizarea pe etape a fiecărui nivel de instruire permite parcurgerea accelerată a acestora în baza traseelor individuale, optimizate pentru fiecare participant în parte.
3. Selectarea resurselor necesare pentru studierea conținuturilor teoretice urmează să fie deosebit de riguroasă, oferind un suport maximal eficient de învățare. Problemele formulate pentru primul nivel de studii – cu fabule și modele matematice clare, ușor detectabile, cu soluții unice, simplu de testat. Complexitatea pe toate palierele urmează să crească treptat, păstrând principiul calității resurselor de învățare.
4. Începând cu etapele nivelului doi legăturile interne între conținuturile studiate devin deosebit de puternice – apar situațiile de existență a soluțiilor multiple, metodelor euristice, soluțiilor intuitive – până a rejecta o soluție diferită sau o idee de rezolvare alta decât cea cunoscută, aceasta urmează să fie riguros cercetată, iar corectitudinea sau incorectitudinea ei – demonstrată.
5. Un puternic motor de învățare este schimbul de experiență între generații. Întâlnirile cu absolvenții olimpici, lecțiile acestora în cadrul activităților, transformă studiul obiectelor și structurilor abstracte într-o experiență reală.
6. Spiritul competiției corecte, elementul principal al menținerii motivației și factorul cheie al creării comunității informatice instituționale, trebuie să persiste la concursurile de toate

nivelele – de la competiția internă, instituțională până la Olimpiada Internațională la Informatică.

7. Metodele pentru comunicarea didactică între mentor și cel (cei) instruiți urmează să fie selectate astfel, încât timpul de așteptare a răspunsului fie cât mai scurt posibil. Prin urmare, în calitate de supliment la metodele de predare în sala de clasă vor veni metodele de predare la distanță, de multe ori – mediate de un sistem informatic. Schemele de comunicare nu mai sunt doar Mentor ↔ Cursant. Sistemele informatice au devenit parte a comunicării și acum schemele de comunicare arată mai complex:
Mentor ↔ [Sistem informatic] ↔ Cursant, Sistemul informatic fiind o componentă care deși se consideră opțională, devine din ce în ce mai de neînlocuit.
8. Modelul dezvoltat integrează componentele instruirii formale, în sala de clasă, cu componentele învățării neformale, prin activități practice planificate, dar conține și momentul informal de instruire – oportunitățile de participare la concursuri reale, în afara orarului oficial de competiții.
9. Modelul dezvoltat permite crearea profilului elevului performer la disciplina Informatică, la etapa selecției pentru pregătirea de concursuri și competiții internaționale. Din acest profil, prin reducere poate fi dedus și profilul potențialului candidat pentru instruirea de performanță la disciplina Informatica.

3. VALIDAREA MODELULUI INTERDISCIPLINAR DE INSTRUIRE PERFORMANTĂ A ELEVILOR ÎN INFORMATICA COMPETIȚIONALĂ – STUDIU DE CAZ

3.1. Specificul spațiului și surselor de date validate

Tradițional, scopul unui experiment pedagogic este testarea, îmbunătățirea, modificarea sau excluderea unor metode, strategii, tehnici de învățare aplicate într-un context specific: învățare formală în sala de clasă sau aula universitară, învățare neformală realizată în instituția de învățământ sau în afara ei, și care se referă la subiecte care nu fac parte din curriculumul general al școlii, învățare informală, sporadică, în baza unor experiențe care nu pot fi aduse încă în mediul educațional, cum, de exemplu, sunt experiențele de utilizare a Inteligenței Artificiale pentru învățare – fiind un instrument conceptual nou, deschis pentru utilizare, abilitățile elevilor și studenților de utilizare ale acestuia au venit din experiențe proprii, recomandări în grupuri și comunități virtuale, și abia mai apoi din partea mediului academic.

De cele mai multe ori cercetările în domeniul științelor Educației se referă la metode și tehnici de îmbunătățire a învățământului formal, realizat în baza unui curriculum la disciplină sau curs. Respectiv, experimentele pedagogice urmează un model de analiză orientat pentru utilizare de scurtă durată – un semestru sau un an de studii. Mai mult, pe parcursul experimentului autorul poate interveni în modelul inițial pentru a-l îmbunătăți sau detalia.

În cazul cercetărilor care vizează spațiul neformal de învățare, iar instruirea de performanță se referă exact la acesta, situația se schimbă radical. Demonstrarea performanței în cadrul disciplinelor școlare este tradițional efectuată prin participarea elevilor în condiții de concurență la concursuri și competiții de diferit nivel. Specificul organizării activităților de concurs nu permite în nici un mod interacțiunea între echipele de mentori și participanți în situații, în care comunicarea ar putea influența rezultatele unui participant sau a unei echipe de participanți. Mai mult, nu poate fi admisă aplicarea unor modele diferite de pregătire pentru participanții la una și aceeași competiție – ar însemna din start defavorizarea unui grup de participanți față de ceilalți, pregătiți de același mentor. În aceste condiții obligația morală a mentorului este să aplice aceeași metodă, același model de instruire pentru toți potențialii participanți la concursuri.

Are loc inversarea etapelor cercetării: modelul de instruire este mai întâi elaborat, aplicat și abia apoi testată eficiența lui prin compararea rezultatelor obținute de toți participanții la concurs. Este un risc suficient de înalt și se încearcă evitarea lui în cazul cercetărilor pedagogice, cu excepția cazurilor când acest mod de validare a rezultatelor instruirii este unicul posibil.

S-a menționat deja că interacțiunea între mentor și instruit în momentele critice (de concurs) este interzisă. Mai mult, este interzisă interacțiunea participanților la concurs cu resurse informaționale altele, decât cele oferite de organizatori. Deci nu se poate interveni, nu se poate corecta sau ajusta un model în procesul validării lui (în timpul concursului). Toate concluziile, analizele, ajustările sunt realizate după afișarea rezultatelor probelor de concurs. Cercetătorul are în aceste situații rolul de observator, iar seturile de date pentru cercetare devin accesibile doar după ce etapa experimentului (competiția) a luat sfârșit.

Așa dar, o analiză a eficienței modelului de instruire aplicat în învățământul de performanță are o serie de particularități, în baza cărora se modifică și modul de validare a rezultatelor;

- Perioada extinsă de realizare. Dacă se urmărește scopul de pregătire a olimpicilor de talie națională / internațională, instruirea poate dura până la finalizarea studiilor, iar rezultatele să apară diferențiat, în funcție de specificul fiecărui instruit;
- Rolul de observator al cercetătorului în perioadele de concurs;
- Modul de implementare a modelului de instruire;
- Imposibilitatea de divizare a participanților da un concurs comun în eșantioane, instruite în condiții diferite.

În contextul prezenței acestor limitări specialiștii recomandă utilizarea unei metode specifice de realizare a experimentului pedagogic - metoda observației științifice.

Metoda își are rădăcinile în lucrările lui Charles Darwin, care a elaborat în baza observațiilor efectuate asupra lumii animale Teoria Evoluției Speciilor [198]. Aplicarea metodei în zona educației pentru studiul dezvoltării cognitive a copiilor a fost propusă și utilizată de Jean Piaget [199], [200]. În educația modernă metoda se aplică pentru analiza comportamentală și analiza progresului elevilor.

Înainte de aplicare o scurtă analiză a metodei propriu zise, a provocărilor lansate de metodă și a punctelor forte. Indiscutabil, printre avantajele metodei se numără studierea fenomenului, acțiunii, procesului în condiții reale. Instrumentele de observare devin din ce în ce mai performante și complexe, colecțiile de date oferite prin observații, respectiv, din ce în ce mai bogate și detaliate. Metoda este flexibilă, în funcție de scopul cercetării poate atât participativă, cât și neparticipativă, în funcție de specificul cercetării. Perioada de aplicare poate fi oricât de lungă. Tocmai aceste aspecte constituie limitările impuse de specificul modelului cercetat, respectiv metoda observației științifice poate fi utilizată pentru argumentarea cercetării.

Metoda observației științifice are mai multe forme, parametri și principii de colectare a datelor, de care s-a ținut cont la realizarea prezentei cercetări. În baza specificului cercetării,

descriș anterior, a fost selectat modelul de observație cu caracter longitudinal, sau observație pe termen lung, care permite identificarea calitativă a modelelor de dezvoltare și evoluție a abilităților, în special, în contextul cercetării efectuate asupra elevilor cu potențial înalt, urmărind cum se dezvoltă competențele instrumentale și interdisciplinare; creativitatea și gândirea algoritmică în timp [201]. Acest model permite și evidențierea zonelor de stagnare a participanților la experiment, gestionarea obstacolelor în condiții reale în intervalele dintre punctele de control – competițiile. Metoda are și un potențial ecologic înalt, deoarece este realizată într-un mediu natural pentru Informatică: clasa, laboratorul de Informatică, mediul de concurs, sistemele informatice – reprezintă ecosistemul pregătirii de performanță la disciplina Informatică. Feed-back-ul este unul continuu, independent de participanți, realizat prin instrumente de evaluare automate, documentat oficial. Suplimentar, pentru integrarea observațiilor în timp au fost elaborate fișe de observare, cu funcție de colectare cumulativă a datelor. Dat fiind specificul domeniului, fișele au fost elaborate în format digital și completate la trecerea fiecărui punct de control (concurs) pe toată perioada instruirii. Un efect colateral al observațiilor, pe termen lung este momentul de predictibilitate a progresului în domeniu după absolvirea instituției de învățământ preuniversitar, atât în planul domeniului profesional, cât și în ceea ce privește progresul profesional personal de carieră.

Aspecte etice. Posibilele probleme din categoria utilizării datelor personale au fost excluse prin utilizarea în exclusivitate a datelor publice, analiza rezultatelor fiind prezentată în mod depersonalizat.

Limitările metodei se referă la cazurile când observarea este efectuată în condiții în care observatul își poate modifica comportamentul, știind că este observat. Situațiile de acest tip sunt excluse, deoarece datorită specificului competițiilor de programare, observația și analiza rezultatelor se face post competiție sau neinvaziv în timpul competiției. În calitate de exemplu se poate aduce monitorul de observație al participanților individuali la Olimpiada Internațională de Informatică (Figura 3.1a) sau pe platforma CodeForces (Figura 3.1b).

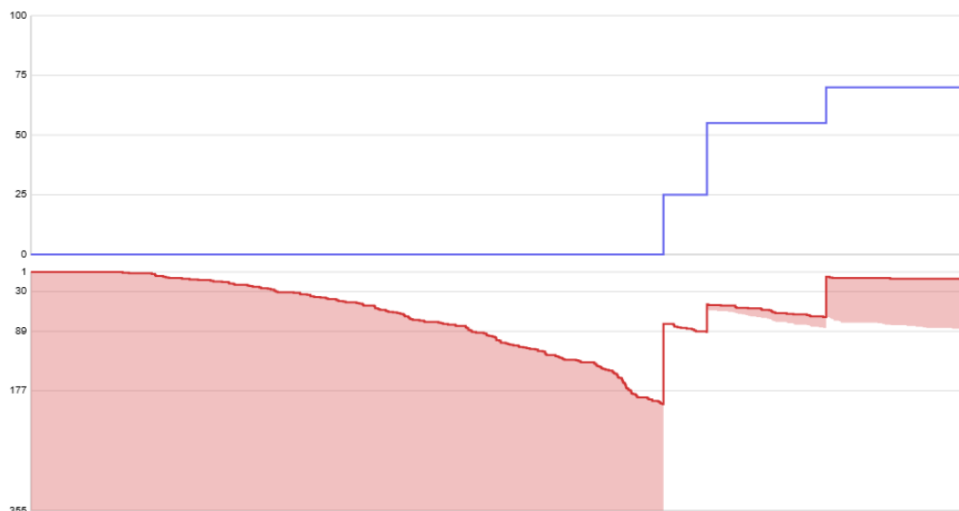
Prelucrarea datelor acumulate de sistemele digitale de monitorizare exclude apariția efectului Hawthorne, permite economisirea timpului și resurselor, garantează corectitudinea datelor colectate.



Bartłomiej
Czarkowski
Poland

	Score	Rank	
Global	341	43	Show
IOI 2021 Day 1	145	53	Show
Distributing Candies	38	59	Show
Keys	37	27	Show
Fountain Parks	70	12	Show
IOI 2021 Day 2	196	50	Show
Mutating DNA	100	1	Show
Dungeons Game	50	72	Show
Bit Shift Registers	46	30	Show

Fountain Parks



Time	Score	Token	Sample	Subtask 1 (5)	Subtask 2 (10)	Subtask 3 (15)	Subtask 4 (20)	Subtask 5 (20)	Subtask 6 (30)
3:22:54	25	No	0	5	0	0	20	0	0
3:28:59	25	No	0	5	0	0	20	0	0

Figura 3.1.a. Fișa de observație a activității participanților

Olimpiadei Internaționale la Informatică (fragment). Fișa conține date despre punctajele acumulate, repartizarea succesului în timp (pe durata competiției), instrumente de selecție a datelor pe probleme separate

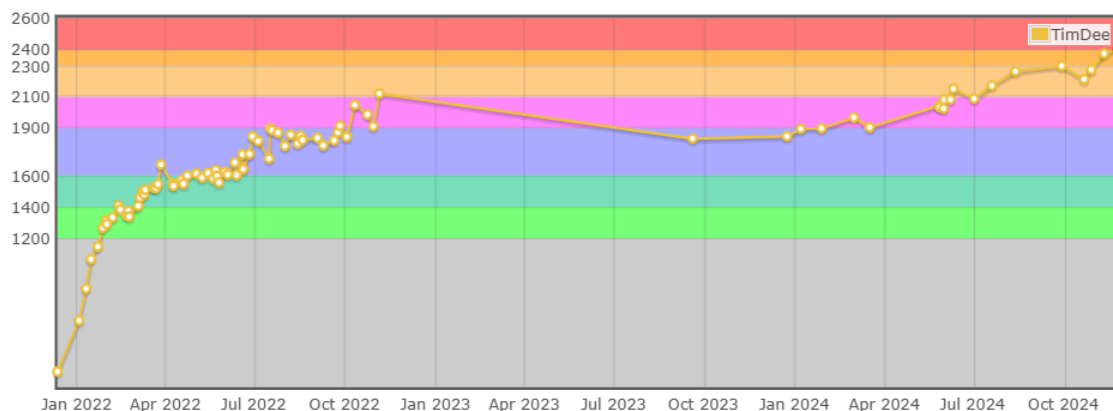


Figura 3.1.b. Fișa de observație a activității celor instruiți pe platforma de pregătire CodeForces. Fișa conține date despre punctajele acumulate în rundele de concurs și dinamica progresului în funcție de complexitatea problemelor

3.2 Analiza comparativă a rezultatelor implementării modelului CRYPTEX de instruire versus metode tradiționale

Studiul de caz a fost realizat în baza datelor despre participanți la edițiile Olimpiadei Naționale în perioada 2017 – 2025, participanții la barajele de calificare a Lotului Olimpic Național, membrii selectați ai echipelor naționale la Olimpiada Internațională de Informatică, perioada 2018 – 2024, Olimpiada Balcanică de Informatică, perioada 2018 – 2024, Olimpiada Europeană de Informatică pentru Juniori, perioada 2018 – 2024, Olimpiada Balcanică de Informatică pentru Juniori, perioada 2018 – 2024.

Primul grup de observații a fost folosit pentru identificarea centrelor de Performanță Instituționale – a acelor instituții de învățământ care pe perioada cercetării au prezentat rezultate sustenabile (un număr total de distincții acumulat, raportat la durata observației în ani mai mare sau egal decât unitatea). Rezultatele totale pe raioane și municipii sunt prezentate în tabelul 3.1

Tabelul 3.1. Repartiția geografică a centrelor de Performanță Informatică

	2017	2018	2019	2022	2023	2024	2025	Total
Chișinău	22	26	23	22	28	25	32	178
Bălți	1	2	2	3	2	4	2	16
Drochia	1	3	3	1	1	0	0	9
Soroca	2	1	2	0	0	1	1	7
Nisporeni	2	1	2	1	0	1	2	9
Tiraspol	2	1	2	1	0	2	1	9
Rezina	0	1	0	0	0	0	0	1
Straseni	1	1	0	1	1	0	0	4
Orhei	1	1	0	0	0	0	0	2
Basarabeasca	0	1	0	0	0	0	0	1
Caușeni	0	1	0	0	0	1	0	2
Hâncești	0	1	0	0	0	0	0	1
Glodeni	0	0	1	1	0	0	0	2
Călărași	1	0	1	1	0	0	0	3
Râșcani	0	0	1	0	0	0	0	1
Criuleni	1	0	1	0	2	2	1	7
Cahul	1	0	1	0	0	1	0	3
Aneni Noi	0	0	0	1	1	0	0	2
UTAG	0	0	0	1	0	1	0	2
Ungheni	0	0	0	2	0	1	0	3
Ialoveni	0	0	0	2	3	1	0	6

Celelalte unități teritorial-administrative nu au obținut pe parcursul perioadei nici una dintre distincțiile oferite de Ministerul Educației și Cercetării prin Consiliul Olimpic la Informatică: zero mențiuni, zero locuri premiante de grad trei, doi și unu.

La următoarea etapă s-a repetat procesul de rafinare a rezultatelor pentru municipiul Chișinău, cu separare pe instituții (Tabelul 3.2):

Tabelul 3.2. Repartiția instituțională a centrelor de Performanță Informatică, Chișinău

	2017	2018	2019	2022	2023	2024	2025	Total
LT Orizont	5	6	9	10	11	9	7	57
CEITI		1	1	2	2	3	1	10
LT M. Eliade	2	3	3	0	0	0	1	9
LT Mihai Viteazul		2	0	0	0	0	0	2
LT Milescu Spătaru		2	1	1	1	1	1	7
LT Petru Rareș	1	1	0	0	0	0	0	2
LAȘM / Aristotel	11	8	7	5	7	5	2	45
LT Prometeu Prim	1	1	1	0	0	0	0	3
LT D. Cantemir		1	1	0	0	0	1	3
LT Iu. Hașdeu		1	0	1	1	0	0	3
LT N. Gogol		1	0	1	0	0	0	2
LT I. Creangă	1	0	0	1	0	0	0	2
LT V. Alecsandri		0	0	1	1	1	0	3
Gimnaziul 42		0	0	0	1	1	0	2
LT S. Haret		0	0	0	1	1	1	3
LT Gaudeamus		0	0	0	1	1	1	3
LT Gh. Asachi		0	0	0	1	0	0	1
LT G. Meniuc		0	0	0	0	1	0	1
LT Ștefan cel Mare		0	0	0	0	1	1	2
LT Berezovschi		0	0	0	0	1	0	1

Generalizând datele din tabelele 3.1 și 3.2 s-a construit lista centrelor de Performanță Informatică instituționale pe întreaga țară (Tabelul 3.3).

Tabel 3.3 Total distincții 2017 - 2025

		Cod
LT Orizont	57	A
CEITI	10	B
LT M. Eliade	9	C
LAȘM / Aristotel	45	D
Bălți	16	E
Drochia	9	F
Soroca	7	G
Nisporeni	9	H
Tiraspol	9	I

Dintre centrele identificate doar în unul (A) se implementează modelul interdisciplinar de instruire al elevilor la disciplina Informatica. Pentru a demonstra pe deplin efectul utilizării acestui model la nivel național datele cantitative (numărul de distincții) vor fi segregate și

calitativ – după tipul distincției, oferind de la 1 punct pentru Mențiunea de Onoare până la 4 puncte pentru Premiul 1.

Pentru a evita analiza unui volum mare de date puțin relevante, în continuare vor fi analizate doar centrele A, B, D, E (Figura 3.2).

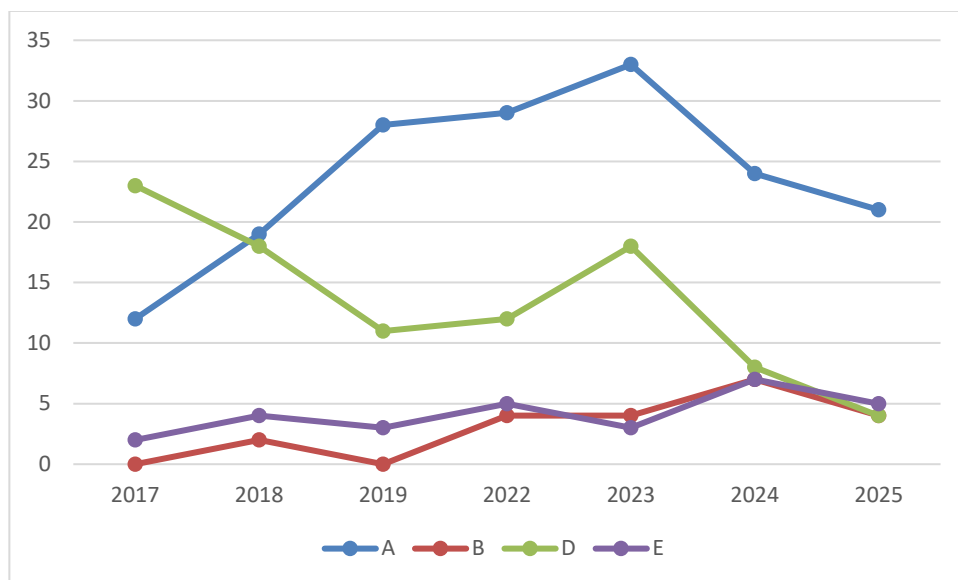


Figura 3.2. Rezultatele calitative ale centrelor Instituționale de Performanță A, B, D, E la Olimpiada Națională de informatică, perioada 2017 - 2025

Se observă poziția dominantă a centrului A pe parcursul întregii perioade de implementare a modelului de instruire interdisciplinar 2018 – 2025, ceea ce confirmă ipoteza dependenței directe a rezultatelor obținute de centrul instituțional de performanță de modelul de instruire utilizat.

Progresul individual al elevilor a fost urmărit în baza fișelor de observație a sistemelor automate de evaluare, utilizate în timpul competițiilor de rang național și internațional. Fișa de sinteză individuală conține pentru fiecare an cu participare activă a elevului calitatea participării la competițiile naționale și internaționale, calculată după principiul:

- Pentru Olimpiada Națională: de la 1 punct pentru Mențiunea de Onoare până la 4 puncte pentru Premiul 1;
- Pentru Olimpiada Balcanică sau Europeană de Informatică: de la 5 puncte pentru Mențiunea de Onoare până la 8 puncte pentru Medalia de Aur;
- Pentru Olimpiada Internațională de Informatică: de la 6 puncte pentru Mențiunea de Onoare până la 9 puncte pentru Medalia de Aur;

Fișele au fost completate pentru toți membrii Echipei Naționale, care au participat la competițiile internaționale în perioada 2017 – 2024. În cele ce urmează pot fi urmărite graficele de progres ale acestora, depersonalizate, cu păstrarea informației despre Centrul Instituțional din

care provin. Astfel, s-a constatat participarea la Olimpiada Internațională la Informatică în perioada 2017 – 2024 a 15 concetanți distincți, dintre care 10 - din centrul instituțional de instruire A, 3 – din centrul instituțional C și 2 din centrul instituțional B (Figura 3.3).



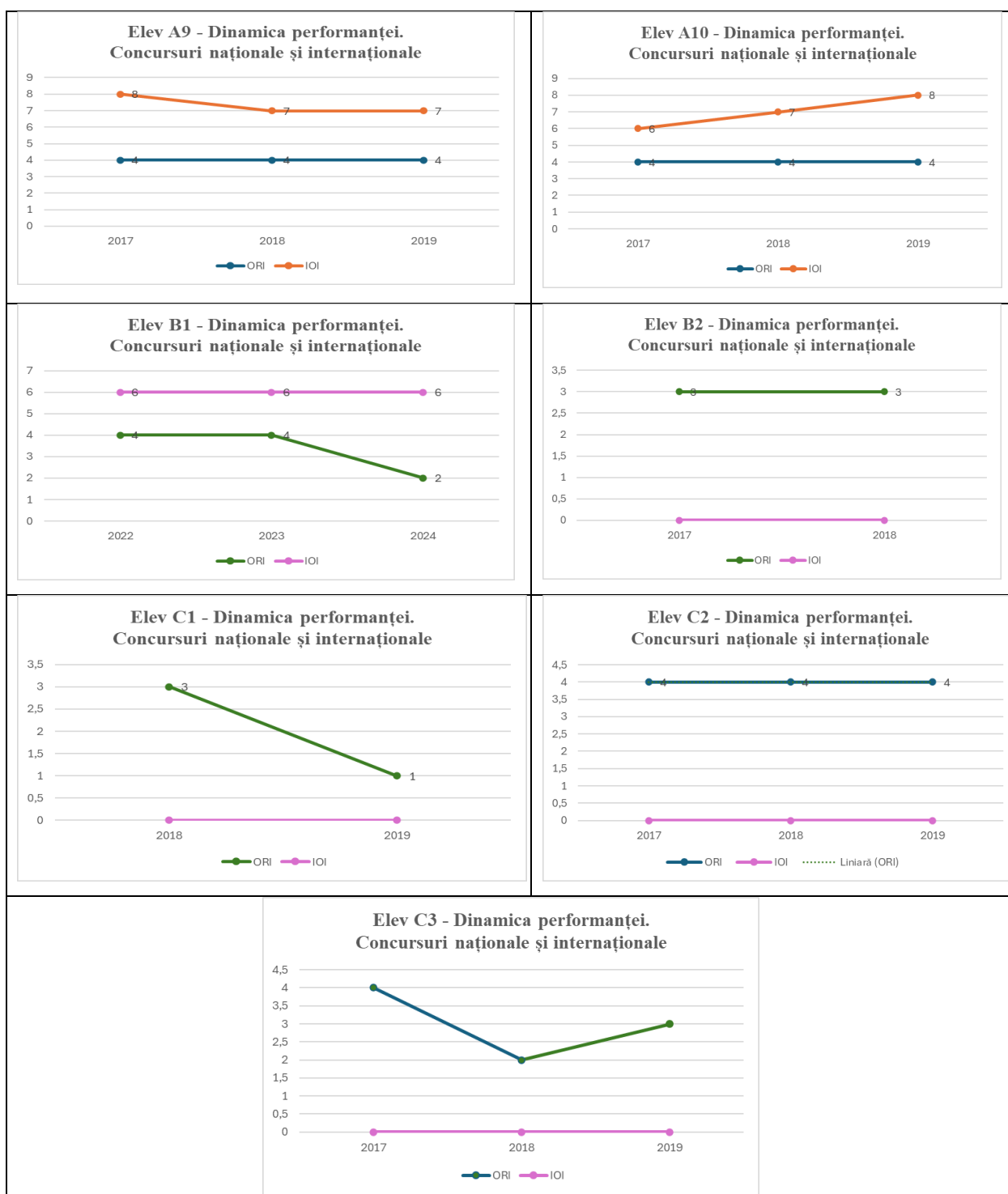


Figura 3.3. Progresul individual al membrilor echipelor olimpice naționale la Informatică perioada 2018 – 2024. Pentru participanții care continuă a căror participări încep până în 2018 sunt incluse seriile complete de rezultate

Analiza rezultatelor individuale permite să fie formulate următoarele constatări:

- 66,66(6)% dintre membrii distincți ai echipelor naționale au fost constituite din elevii centrului de instruire A; 13,3(3)% au fost constituite din elevii centrului de instruire D; 20% au fost constituite din elevii centrului de instruire C;

- creșterea sau menținerea performanțelor în cadrul Olimpiadelor Internaționale pe parcursul întregii perioade de participare - pentru 60% dintre participanții la Olimpiada Internațională. Dintre cei 9 participanți cu această performanță 8 sunt parte a centrului de instruire A, 1 – a centrului de instruire B;
- descreșterea calității rezultatelor la Olimpiada Internațională la Informatică se atestă la 1 reprezentant (6,66(6)% din numărul total cercetat) – al centrului de Instruire A. În contextul abordării interdisciplinare urmează să fie menționat că în aceeași perioadă, în care se atestă micșorarea performanței informatice, același elev a obținut performanțe echivalente și la Olimpiada Internațională la Matematică; De remarcat că încă cel puțin trei dintre participanți din această categorie au înregistrat succese remarcabile la Olimpiada Internațională de Matematică și Olimpiada Internațională de Fizică în perioada analizată;
- lipsa totală a rezultatelor pe parcursul participărilor la Olimpiadele Internaționale se atestă la 4 participanți, dintre care 1 participant din cadrul centrului de instruire A, 1 participant din cadrul centrului de instruire B, 3 participanți din cadrul centrului de instruire C.

Rezultatele competițiilor informatice sunt rezultate care nu au nevoie de o validare suplimentară. Cu toate acestea, analiza dinamicii progresului performanței în timp confirmă calitativ rezultatele competiției atestând o creștere continuă sau menținere stabilă a performanței pentru cel puțin 80% dintre elevii centrului instituțional A. Pentru reprezentanții centrului de instruire D acest indicator este de 50%, dar acest rezultat este mai puțin relevant din cauza numărului mic de participanți (2 în total).

Prin urmare, modelul de instruire interdisciplinar, în baza căruia au fost și sunt instruiți elevii centrului instituțional de performanță A și-a demonstrat atât statutul de lider la nivel național cât și un potențial real de competitivitate cu modelele de instruire a liderilor internaționali în performanța informatică.

3.3. Analiza post – instrucțională

Instruirea de performanță în baza modelului CRYPTEX nu doar permite formarea unor elevi cu capacități extraordinare în domeniul Informaticii. O analiză a parcursului post liceal pentru grupul de elevi cercetat în paragraful precedent demonstrează capacitatea lor de a păstra tendința de creștere a potențialului informatic și după finalizarea studiilor preuniversitare.

În continuare sunt prezentate rezultatele analizei după următoarele criterii de creștere a potențialului informatic, în aspect competitiv:

1. Universitatea la care și-au continuat studiile după absolvirea liceului;

2. Participarea la competițiile internaționale studentești de programare competitivă;
3. Participarea oficială sau în calitate de voluntari în activitatea Consiliului Național Olimpic la disciplina Informatică;
4. Participarea oficială sau în calitate de voluntari în organizarea competițiilor naționale de informatică;
5. Participarea oficială sau în calitate de voluntari în organizarea competițiilor internaționale de informatică;
6. Elaborarea de probleme, teste și soluții pentru concursuri de programare naționale și / sau internaționale.

Statisticile au fost colectate prin intermediul formularelor Google, online, de la absolvenții anilor 2020 – 2024 ai liceului Orizont, incluși în procesul de observație descris anterior. A fost păstrată atât codificarea, utilizată anterior, cât și componența grupului observat. Pentru a depersonaliza datele despre universitățile la care au fost acceptați elevii performeri s-a utilizat ratingul internațional *QS World University Rankings 2025: Top global universities*, [202] cu divizările pe intervale: top 10 universități din lume, top 50 universități din lume, top 250 universități din lume (Figura 3.4).

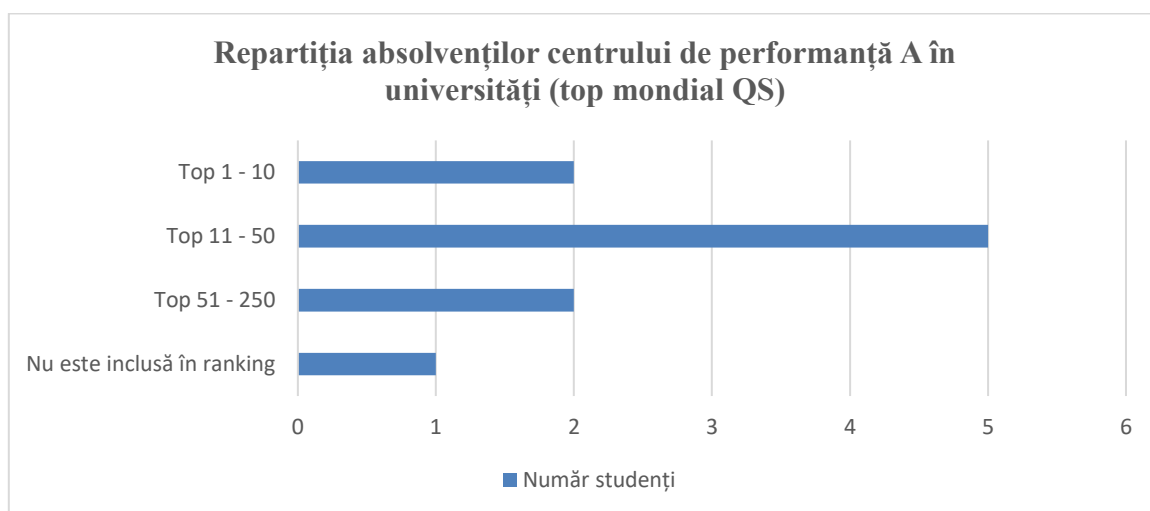


Figura 3.4. Ratingul mondial al universităților la care și-au continuat studiile absolvenții centrului de performanță A, olimpici internaționali la disciplina Informatica

Opt din zece intervievați au continuat să participe la concursurile de programare studentești tip ICPC (International College Programming Contests) în perioada 2020 – 2024 în calitate de competitori, demonstrând în continuare creșterea abilităților de programare competitivă în activități practice. În același timp, în diferite moduri, absolvenții s-au manifestat în calitate de organizatori sau membri a echipelor de organizare a concursurilor naționale și internaționale. Principalele activități competiționale de performanță realizate cu sprijinul lor, pe parcursul

perioadei 2020 – 2024 au fost apariția în calitate de membru al Consiliului Olimpic la disciplina Informatică, participarea în organizarea Olimpiadei Naționale de Informatică, Comisia Științifică a concursului „The Game of Code”, realizarea rundelor practice pentru grupele de programatori-începători pe platforma CodeForces.

Tabelul 3.4 descrie modurile de implicare a fiecărui absolvent în activități administrative și de cercetare aferente concursurilor de programare. În calitate de membri ai comitetelor științifice pentru competiții internaționale, absolvenții centrului de performanță A apar la *Olimpiada Europeană de Informatică pentru Juniori*, ediția 2024, *Olimpiada Europeană de Informatică pentru fete*, ediția 2024, *International Advanced Tournament in Informatics* ediția 2025.

Tabelul 3.4. Implicarea absolvenților în activități creative și de cercetare asociate concursurilor de programare

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Participarea în activitatea Consiliului Olimpic la disciplina Informatică	☐	☐	☐	☒	☒	☒	☒	☐	☒	☒
Participarea în organizarea Olimpiadei Naționale de Informatică	☒	☒	☐	☒	☒	☒	☒	☐	☒	☒
Participarea în organizarea Concursurilor Internaționale de Informatică	☒	☐	☐	☒	☒	☒	☒	☐	☒	☐
Elaborarea seturilor de probleme, soluții și teste pentru concursuri	☒	☒	☐	☒	☒	☒	☒	☐	☒	☒

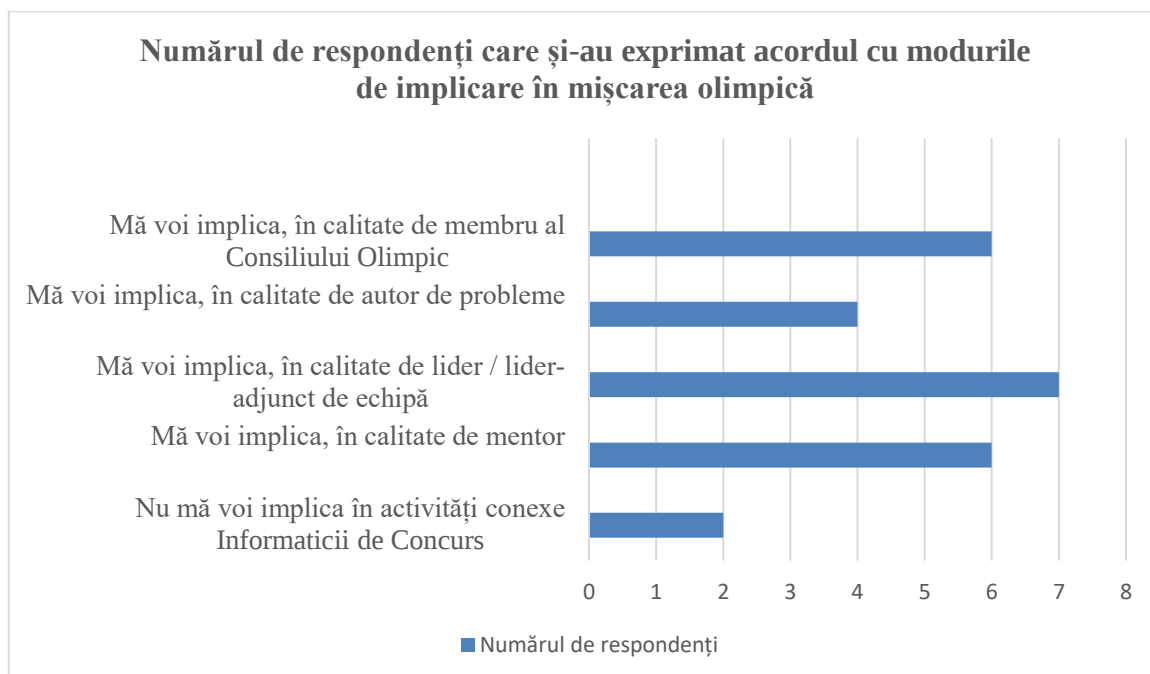


Figura 3.5. Preferințele de implicare, pe tipuri de activități, a absolvenților centrului de performanță A pentru în activitatea Consiliului Olimpic la disciplina Informatica

Activitatea absolvenților în domeniul instruirii de performanță la disciplina Informatică se extinde și pe domenii conexe pedagogiei și psihologiei școlare: doar din grupul analizat anterior

trei membri au apărut în perioada 2020 – 2024 în calitate de lideri sau lideri-adjuncți de echipă la diverse concursuri internaționale, printre care: *Romanian Masters in Informatics*, *Olimpiada Balcanică la Informatică pentru Juniori*, *Olimpiada Europeană la Informatică pentru Juniori*.

Mai mult decât jumătate dintre absolvenții instruiți în baza modelului CRYPTEX își rezervă în activitățile profesionale ulterioare, cel puțin pentru următorii trei – cinci ani resurse de timp și efort pentru activitățile din zona Informaticii Competitive. Un ultim bloc de întrebări din Chestionarul propus demonstrează acordul a până la 70% dintre respondenți să se implice în diverse zone ale activității de cercetare sau administrative, conexe organizării concursurilor de programare, Figura 3.5.

3.4. Concluzii Capitolul 3

Calitatea modelului interdisciplinar de instruire de performanță la disciplina Informatică a fost confirmată în condiții de competiții reale, naționale și internaționale, oficiale și neoficiale pe parcursul a două perioade integrale de instruire 2017 – 2019 și 2021 – 2024, demonstrând stabilitatea rezultatelor dar și un feedback pozitiv din partea celor instruiți [66, pp. 59-72] Studiul de caz, realizat pe două nivele, în baza datelor statistice, colectate atât la nivel național, cât și internațional a permis formularea următoarelor concluzii:

1. În Republica Moldova există mai multe centre instituționale de performanță informatică, cinci dintre care sunt localizate în municipiul Chișinău, celelalte – în municipiile Bălți, Tiraspol, raioanele Drochia, Nisporeni și Soroca. Principalele centre de performanță, după numărul și calitatea rezultatelor la Olimpiada Națională la Informatică sunt centrele liceelor teoretice Orizont, Aristotel, Mircea Eliade, CEITI.
2. Utilizarea experimentală a modelului propus în unul dintre centrele enumerate, pe parcursul unui interval considerabil de timp, a permis creșterea rapidă, atât numerică, cât și calitativă a rezultatelor acestuia pe plan național și internațional.
3. Metodele tradiționale, utilizate pentru validarea rezultatelor experimentelor pedagogice efectuate nu pot fi utilizate în cazul cercetării curente din mai multe motive, printre care: durata lungă a experimentului, specificul selecției inițiale a participanților, nivelul înalt de automotivare a participanților, condițiile externe de realizare a testelor, nivelul înalt de complexitate al testelor.
4. O metoda acceptabilă de validare a modelului elaborat, ecologică și neinvazivă este metoda observației. Utilizarea ei este facilitată și de instrumentele automate de colectare a datelor statistice, integrate în sistemele de evaluare ale Olimpiadei Naționale și a Olimpiadelor Internaționale.

5. Studiul de caz a demonstrat eficiența modelului de instruire elaborat, atât cantitativ, analizând rezultatele fiecărui centru instituțional la Olimpiada Națională, pe ani, pentru perioada 2017 – 2025, cât și calitativ, prin analiza individuală a progresului rezultatelor fiecărui membru al echipei Naționale la Informatică, atât în plan național, cât și în plan internațional, în timp, pe ani. Perioada de analiză este întreruptă în anii 2020 și 2021 pe motiv că în perioada pandemică nu au fost organizate Olimpiadele naționale la Informatică (2020, 2021), respectiv nici barajele pentru selecția echipei naționale, în consecință nu a fost posibilă participarea la ediția din anul 2020 a Olimpiadei Internaționale.
6. În urma analizei datelor statistice s-a constatat, că circa 80% din numărul participanților la Olimpiadele Internaționale, care demonstrează un progres permanent, sau rezultate stabilizate la nivel internațional, au fost formați în conformitate cu modelul interdisciplinar de instruire de performanță la Informatică, inclusiv fiind exploatată maximal posibil componenta de antrenamente periodice prin participare la concursuri internaționale, neofici-alizate de Consiliul Olimpic la Informatică.
7. Același studiu de caz a demonstrat, că modelele de instruire de performanță, exploatate în alte centre instituționale:
 - a. sunt orientate preponderent către obținerea rezultatelor la nivel național;
 - b. obțin deseori rezultate sporadice, se confruntă cu lipsa resurselor pentru susținerea performanței;
 - c. nu folosesc oportunitățile de participare la concursurile neoficiale de pregătire orga-nizate online sau onsite;
8. Cele menționate demonstrează necesitatea preluării de către alte instituții de învățământ a modelului interdisciplinar de instruire elaborat, inclusiv cu sesiuni de formare a cadrelor didactice, care pot fi organizate în centrele universitare de formare continuă. O asemenea practică va duce la revigorarea spiritului de concurență între centrele de performanță, ceea ce în consecință va avea drept rezultat creșterea calității rezultatelor, obținute de olimpicii din Republica Moldova pe arena internațională.
9. Modelul de instruire elaborat formează nu doar competențe profesionale, dar și o atitu-dine corectă față de comunitatea olimpică. Implicarea absolvenților în activitatea Consi-liului Olimpic, organizarea concursurilor, acordul / intenția de a contribui la dezvoltarea comunității olimpice informatice demonstrează și corectitudinea modelului comporta-mental, dezvoltat pe parcursul anilor în cadrul instruirii de performanță la disciplina In-formatica.

CONCLUZII GENERALE ȘI RECOMANDĂRI

Modelul pentru instruirea de performanță la Informatică a apărut din necesitatea de a rezolva problema lipsei rezultatelor de valoare a Republicii Moldova pe arena informatică internațională. Altfel spus e un model, destinația căruia este antrenarea participanților la competițiile internaționale de Informatică pentru obținerea rezultatelor de valoare, care să contribuie la promovarea imaginii Republicii Moldova.

Analiza insucceselor pe plan internațional în perioada 2008 - 2013 a adus în prim plan discrepanța crescândă între Curriculumul Național la Informatică și Syllabus-ul Internațional de pregătire pentru concursuri, insuficiența de ore, alocate studiului Informaticii ca știință, lipsa instituțiilor de profil specializate, care să ofere un nivel de pregătire suficient de înalt pentru a face față provocărilor Informaticii moderne.

Prima etapă a cercetărilor s-a desfășurat în perioada 2011 -2013. Intervalul de timp a coincis cu perioada de implementare a Curriculumului modernizat dar și cu perioada actualizării Syllabus-ului Internațional. În numeroase publicații s-a semnalat necesitatea introducerii unui Curriculum de Performanță la Informatică. În lipsa unui suport metodologic specific, și datorită posibilității de a valorifica potențialul elevilor din cadrul liceului Orizont, cercetarea a început în forma unor experiențe practice de pregătire în baza Syllabus-ului Internațional, la care s-a adăugat posibilitatea de a valorifica resursele platformelor online de programare competitivă - pentru rezolvarea de probleme, exersare continuă, participare la concursuri virtuale.

Modificările și optimizările de conținuturi pentru învățare au continuat până în 2016 – 2017, când, pe de o parte, s-a stabilizat conținutul Syllabus-ului Internațional, pe de alta – au (re)apărut primele rezultate semnificative ale informaticienilor din Republica Moldova – în noua serie de distincții s-au adăugat doar la IOI 6 medalii, fără a ține cont de rezultatele la alte competiții internaționale.

Modelul a fost testat ulterior pe elevii generațiilor 2017 – 2019 și 2021 – 2024, demonstrând sustenabilitatea sa și posibilitatea de aplicare în diverse condiții, inclusiv pandemice.

Constatările, pentru momentul actual sunt următoarele:

1. Problema de a readuce Republica Moldova în lista țărilor vizibile internațional în contextul Informaticii Competitive a fost rezolvată. Pe parcursul a zece ani, reprezentanții Republicii Moldova la Olimpiadele Internaționale au demonstrat performanțe constante. Performanțele aparțin membrilor echipei naționale, care au fost pregătiți de competiții conform modelului interdisciplinar de instruire pentru performanța Informatică

2. Obiectivele formulate au fost realizate integral: au fost identificate centrele naționale de instruire pentru performanța informatică, a fost analizat modelul național actual de instruire pentru performanță informatică; au fost cercetate modelele de instruire în România, Bulgaria (țări vecine); A fost analizată structura probelor de concurs în cadrul Olimpiadei Naționale și Internaționale de Informatică; au fost determinate diferențele dintre metodele de formare și dezvoltare a competențelor informatice de performanță la nivel internațional și nivel național; a fost elaborat un model național de instruire adaptat pentru performanța la nivel internațional, eficiența căruia a fost validată în perioadele 2017 – 2019 și 2021 - 2024;
3. Specificul mediului în care este efectuată cercetarea implică o abordare specifică în interpretarea rezultatelor. Durata extinsă a cercetării, combinată cu condițiile speciale în perioadele de control, evaluarea externă a rezultatelor prin scoruri oferite în competiții, imposibilitatea urmăririi directe în condiții de concurs, dar și generarea de către sistemele de evaluare a fișelor de observație digitale a trecut procesul de validare statistică în zona observării neinvazive și a studiului de caz în baza unor rezultate a căror veridicitate nu poate fi contestată – a scorurilor obținute la competițiile naționale și internaționale de fiecare dintre cei observați: 15 din 16 distincții obținute la Olimpiada Internațională de Informatică aparțin elevilor centrului instituțional de performanță informatică al liceului Orizont. Este un indicator de 94% din numărul total de distincții obținute.
4. Rezultatele obținute în centrul instituțional de performanță informatică al liceului Orizont nu au putut fi replicate în alte instituții de învățământ, chiar dacă acestea prezintă rezultate de performanță constante și stabile la nivel național.
5. Numărul dominant de distincții la competițiile naționale, împreună cu datele statistice ale rezultatelor participării la competițiile internaționale demonstrează ferm eficiența cercetării, care, pe parcursul mai multor etape ale sale, a avut în calitate de rezultat elaborarea modelului interdisciplinar de instruire de performanță informatică și implementarea acestuia în cadrul liceului Orizont.
6. Modelul de instruire elaborat este specific și prin un efect secundar, care se observă în urma analizei datelor statistice colectate în baza fișelor individuale de concurs: elevii instruiți după modelul interdisciplinar păstrează tendința către progresul rezultatelor la competiții și după absolvire – 8 din 10 grafice indică creșterea sau menținerea nivelului de performanță la o extrapolare pentru perioadele ulterioare.
7. Preluarea și implementarea modelului interdisciplinar în alte centre instituționale de performanță poate genera creșterea nivelului de concurență pentru selecția în lotul și ulterior

pentru selecția în echipa națională pentru participarea la Olimpiada Internațională la Informatică dar și la olimpiadele regionale, cum ar fi Olimpiada Europeană pentru Fete și Olimpiada Balcanică la Informatică. Creșterea concurenței este un instrument de menținere și îmbunătățire a rezultatelor Republicii Moldova la competițiile informatice pe plan internațional.

8. Menținerea modelului propus în stare funcțională în cadrul instituțiilor de învățământ preuniversitare necesită eforturi majore și din partea mentorilor și din partea administrației instituției. Mentorul este pus în situația de a învăța permanent, fără o zonă de confort prestabilit, să rezolve noi și noi probleme, a căror nivel de complexitate depășește mult nivelul de complexitate a cursurilor de specialitate studiate la facultate sau în cadrul formării continue. Nu oricine este gata să-și asume un asemenea grad de responsabilitate fără susținerea unei comunități profesionale competente, sau a unei motivații puternice.

Recomandări:

1. O vizibilitate mai mare a rezultatelor cercetării poate fi obținută prin lansarea unui portal web național, care să ofere tuturor celor interesați de programarea competitivă tot setul de resurse necesare în contextul utilizării modelului interdisciplinar de instruire pentru performanța informatică, cel puțin: resurse teoretice, intrări pe platforme de exersare și evaluare automată, informații curente, oportunități de participare la competiții, contacte ale mentorilor și olimpicilor absolvenți.
2. Pentru diminuarea presiunii pe centrele instituționale de performanță sunt necesare acțiuni coordonate pe câteva direcții:
 - a. Revizuirea și modernizarea Curriculumului Național la disciplina Informatică în sensul readucerii disciplinei în zona de științe fundamentale – Computer Science, separând modulele tehnologice într-o disciplină separată – Tehnologii Informaționale, alocarea orelor la disciplina Informatică în clasele 6 – 7, introducerea bazelor programării ca modul obligatoriu.
 - b. Modernizarea cursurilor universitare în domeniul algoritmicii, structurilor de date, arhitecturii calculatoarelor, introducerea cursurilor de analiză a interviurilor de angajare la companiile IT internaționale – peste 50% din întrebările de profil în aceste interviuri sunt probleme de la competițiile informatice. Crearea echipelor studențești pentru participarea la concursurile de programare între universități.
 - c. Participarea oficială, la nivel național, la concursuri internaționale cu desfășurare permanentă sau pe runde, cum este Olimpiada Internațională la Informatică pe

Echipe, unde pot fi antrenați, pentru etapa națională un număr nelimitat de participanți.

- d. Crearea centrelor de performanță informatică universitare, unde ar fi implicați masteranzii și doctoranzii pe domeniile aferente programării competiționale. Lansarea în baza acestor centre a concursurilor naționale de programare, în diverse formate, cu acces deschis.
3. Revizuirea periodică a modelului elaborat în scopul actualizării conținuturilor, corelării cu Syllabus-ul internațional și Curriculumul Național la Informatică, adăugării și modernizării resurselor pentru învățare, inclusiv a resurselor interactive și inteligente.
4. Dezvoltarea, în baza profilului elevului performer la disciplina Informatica, a profilului candidatului pentru inițierea timpurie în bazele Informaticii Competitive, iar ulterior – pentru instruirea în baza modelului CRYPTEX.

BIBLIOGRAFIE

- [1] TOFFLER, A. *Future shock*. New York: Bantam Books, 1970.
- [2] WIENER, N. *Cybernetics Or Control and Communication in the Animal and the Machine*. USA: MIT Press, 1948, 1961.
- [3] CERUZZI, P. E. *Computing: A Concise History*. London: MIT Press, 2012.
- [4] HAROON, A. *Cambridge Computing*. London, UK: Third Millennium Publishing Limited, 2013.
- [5] MAHONEY, M. S. The History of Computing in the History of Technology. In: *Annals of the History of Computing*, 1998, vol. 10, nr. 2, pp. 113-125.
- [6] *UNESCO Records of the General Conference*. In: *Resolutions*, Paris, 1987.
- [7] GREMALSCHI, A., PRISACARU, A., **CORLAT, S.** Olympiads in Informatics in Republic of Moldova. In: *Olympiads in Informatics*, 2016, Kazan.
- [8] BROWN, I. *Regulation and the Internet of Things*. Oxford Internet Institute, Oxford, 2015.
- [9] *The Societal Impact of the Internet of Things*. British Computer Society, Oxford Internet Institute, 2013.
- [10] ANDREEV, V. Pedagogical Factors Stimulating the Self-Development of Students, Multi-Dimensional Thinking in Terms of Subject-Oriented Teaching. In: *International Education Studies*, 2014, vol. 7, nr. 7, pp. 63-68.
- [11] Strategia Națională de edificare a Societății Informaționale - Moldova Electronică. In: *Monitorul Oficial al Republicii Moldova*, 2005, nr. 46-50 art. nr. 336.
- [12] Strategia națională de dezvoltare Moldova 2030. Parlamentul Republicii Moldova. In: *Monitorul Oficial al Republicii Moldova*, 2021.
- [13] Agenția de Guvernare Electronică. Strategia de transformare digitală 2023 - 2030, 2023. [Interactiv]. Available: https://www.egov.md/sites/default/files/document/attachments/strategia_de_transformare_digitala_2023-2030.pdf. [Accesat 03 aprilie 2025].
- [14] *Strategia de transformare digitală 2023 - 2030*. Agenția de Guvernare Electronică. 2023. Disponibil: https://www.egov.md/sites/default/files/document/attachments/strategia_de_transformare_digitala_2023-2030.pdf. [citat 03.04.2025].
- [15] Codul Educației al Republicii Moldova. In: *Monitorul Oficial al Republicii Moldova*, 2014.
- [16] BOYER, C. B., UTA, M. *A History of Mathematics*. Hoboken: Wiley & sons, 2013.
- [17] IFRAH, G. *The Universal History of Computing: From the Abacus to the Quantum Computer*. New York, NY: John Wiley & Sons, Inc, 2001.

- [18] SEDGWICK, W. T., TYLER, H. *A Short History Of Science*. New York, US: The Macmillan Company, 1939.
- [19] BOOLE, G. *The Mathematical Analysis of Logic*. London: Henderson & Spalding, 1847.
- [20] BOOLE, G. *An Investigation of the Laws of Thought: on Which are Founded the Mathematical Theories of Logic and Probabilities*. New York: Cosimo, Inc., 1854 (2007).
- [21] HOARE, T. QuickSort. In: *The Computer Journal*, 1961, vol. 5, nr. 1, pp. 10-16.
- [22] KNUTH, D. *The Art of Computer Programming: Vol. 1: Fundamental Algorithms*. 3rd Edition, Boston: Addison-wesley Professional, 1997.
- [23] KNUTH, D. *The Art of Computer Programming, Vol. 2 / Seminumerical algorithms*. Massachusetts: ADDISON-WESLEY, 1997.
- [24] KNUTH, D. *The art of computer programming. Vol. 3 / Sorting and Searching*. Massachusetts: ADDISON-WESLEY, 1997.
- [25] KNUTH, D. *Art of Computer Programming, Vol. 4, Fascicle 4: Generating All Trees- History of Combinatorial Generation*. Massachusetts: AddisonWesley Professional, 2006.
- [26] KNUTH, D. *The art of computer programming Vol. 4, Fascicle 6 Satisfiability*. Pearson Education, Inc., 2015.
- [27] КРИСТОФИДЕС, Н. *Теория графов. Алгоритмический подход*. Москва: Мир, 1978.
- [28] GIBBONS, A. *Algorithmic graph theory*, Addison Wesley, 1999.
- [29] GRUS, J. *Data Science from Scratch*. Printed in the United States of America: OReilly, 2015.
- [30] WILLIAMSON, G. G. *Combinatorics for Computer Science*. Ontario: General Publishing Company, Canada, 1985.
- [31] PREPARATA, F., SHAMOS, M. *Computational Geometry. An Introduction*. New York: Springer-Verlag, 1989.
- [32] **CORLAT, S.** Aplicații ale Geometriei computaționale în rezolvarea problemelor de Informatică. In: *Delta*, 2007, nr. 4, pp. 24-27.
- [33] VINCE, J. *Mathematics for Computer Graphics*. London: Springer, 2010.
- [34] GREMALSCHI, A. **CORLAT, S.** BRAICOV, A. *Informatica. Manual pentru clasa a XII-a*. Chișinău: Știința, 2025, p. 202.
- [35] GREMALSCHI, A., **CORLAT, S.**, BRAICOV, A. *Informatica Manual pentru Clasa a XII-a*. Chișinău: Știința, 2010.

- [36] BRAIN, C. *The Alignment Problem. Mashine Learning and Human Values*. New York: W.W. Norton & Company, 2020.
- [37] ШТЕЙНГАУЗ, Г. *Задачи и размышления*. Москва: Мир, 1974.
- [38] *Facultatea de Matematică și Informatică. USM* [online]. Disponibil: <https://fmi.usm.md/scurt-istoric/>. [citat 31 03 2025].
- [39] ЗВЕНИГОРОДСКИЙ, Г. Заочная школа программирования. Законы программирования. Урок 1. In: *Квант*, 1979, nr. 9.
- [40] КАСЬЯНОВ, В. Заочная школа программирования. Законы программирования. Урок 20. *Квант*, 1982, nr. 4.
- [41] **CORLAT, S., KARLSSON, G.** *Metodologia Utilizării Tehnologiilor Informaționale și de Comunicație în învățământul superior*. Chișinău: F.E.P. Tipografia Centrală, 2011.
- [42] ЕРШОВ, А. *Основы информатики и вычислительной техники: часть I*. Москва: Просвещение, 1985.
- [43] ЕРШОВ, А. *Основы информатики и вычислительной техники: часть II*. Москва: ПРОСВЕЩЕНИЕ, 1886.
- [44] BOSTAN, G. *Culegere de probleme de informatică*. Chișinău: Lumina, 1996.
- [45] GREMALSCHI, A., MOCANU, I., SPINEI, I. *Informatica Limbajul Pascal Manual pentru clasele IX - XI*. Chișinău: Știința, 1999.
- [46] **CORLAT, S., SINGER, B.** *Totul despre calculatoare pentru cei mici, parinti si bunici*. Chișinău: Prut Internațional, 1996.
- [47] **CORLAT S., IVANOV, L.** *Calcul numeric. Curs de lecții*. Chișinău: CCRE Presa, 2004.
- [48] WIRTH, N. *The Programming Language Pascal (Revised Report)*. ETH Zürich, Zürich, 1973.
- [49] LECARME O., DESJARDINS, P. More comments on the programming language Pascal. In: *Acta Informatica*, vol. 4, 1975, nr. 3, p. 231-243.
- [50] *Curriculum Național Informatică*, Chișinău: Liceum, 2020.
- [51] *Curriculum național : Clasele 10-12*, Chișinău: Lyceum, 2020.
- [52] **CORLAT, S.** Curriculumul Național - 2019 la Informatică: modernizare structurală și de conținut In: *Didactica Pro*, 2019, nr. 4-5(116-117), pp. 53 - 56.
- [53] ADĂSCĂLIȚEI, A. *Instruirea asistată de calculator*. Iași: Polirom, 2007.
- [54] AGUSTI, M., VELASCO, M., SERRANO, M. E-Learning: Psycho-pedagogical utility, usability and accessibility criteria from a learner centred perspective. In: *Handbook of*

research on e-learning standards and interoperability; Frameworks and issues. Hershey, 2011, pp. 419-434.

- [55] CLARK, R. C., MAYER, R. E. *E-Learning and the science of instruction*. San Francisco: Pfeiffer, 2011.
- [56] *Programe școlare în vigoare și Repere Metodologice*. [online]. Disponibil: <https://rocnee.eu/index.php/dcee-oriz/curriculum-oriz/planuri-cadru-actuale>. [citat 01.04.2025].
- [57] *Министерство на образованието и науката*. [online]. Disponibil: <https://www.mon.bg/obshto-obrazovanie/uchebni-planove-i-programi-2/uchebni-programi/>. [citat 01.04.2025].
- [58] *The Art of Problem Solving* [online]. Disponibil: https://artofproblemsolving.com/wiki/index.php/IMO_Problems_and_Solutions. [citat 01 04 2025].
- [59] *International Olympiad in Informatics - Statistics 2025*. [online]. Disponibil: <https://stats.ioinformatics.org/olympiads/>. [citat 01 04 2025].
- [60] GREMALSCHI, A., **CORLAT, S.** Instruirea de performanță la Informatică în contextul curriculumului general. *Didactica Pro*, 2011, nr. 5-6 (69-70).
- [61] LUPU, I., GREMALSCHI, A., **CORLAT, S.** Proiectarea curriculară în învățământul de performanță la Informatică. In: *Acta et Commentationes (Științe ale Educației)*, 2014, nr. 2(5).
- [62] GREMALSCHI, A., **CORLAT, S.**, PRISACARU, A. Olympiads of Informatics in Republic of Moldova. In: *Olympiads in Informatics*, 2016, vol. 10.
- [63] *EJOI 2024* [online]. Disponibil: <https://ejoi2024.gov.md/>. [citat 01.04.2025].
- [64] *ANCE*, 2023. [online]. Disponibil: https://ance.gov.md/sites/default/files/ol23_informatica_probleme.pdf. [citat 04.04.2025].
- [65] THOMPSON, K. *Interdisciplinarity: History, Theory and Practice*. Detroit: Wayne State University Press, 1990.
- [66] CHIRIAC, L. *Evaluarea procesului de studiere a științelor reale și ale naturii din perspectiva inter/transdisciplinarității (concept STEAM)*. Chișinău: Tipografia Centrală, 2020.
- [67] NEWMAN, M. *Computational Physics*. Michigan: University of Michigan, 2013.

- [68] SEDGEWICK, R. *Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms*. New York: Addison Wesley, 2001.
- [69] JENSEN, F. *Introduction to Computational Chemistry*. Chichester, West Sussex: John Wiley & Sons, Ltd, 2017.
- [70] ACHUTHSANKAR N. *Computational Biology & Bioinformatics: A Gentle Overview*. Communications of the Computer Society of India, 2007.
- [71] КИРЮХИН В. М. *Методика проведения и подготовки к участию в олимпиадах по Информатике*. Москва: Бином, 2011.
- [72] DIKS, K., KUBICA M., STENCEL K. Polish Olympiad in Informatics – 14 Years of Experience. In: *Olympiads in Informatics*. 2007, vol. 1, pp. 50 - 57.
- [73] ERACLEOUS, P., PAVLIKAS, P., TTOFARI, A., CHARALAMPOUS, A. Cyprus Olympiad in Informatics. In: *Olympiads in Informatics*. 2019, vol. 13, pp. 193–200. <https://doi.org/10.15388/IOI.2019.12>.
- [74] VLISNOVITZ, M., HORVÁTH, G. Trends in Teaching Programming in Schools in Hungary. In: *Olympiads in Informatics*. 2023. vol. 17, pp. 121 - 130.
DOI: 10.15388/ioi.2023.09
- [75] PAVEL, B. PANKOV, S. Olympiads in Informatics in Kyrgyzstan. In: *Olympiads in Informatics*, 2022, vol. 16, pp. 125-133.
- [76] LAAKSONEN, A. *What is the Competitive Programming Curriculum?* 2022, vol. 16, pp. 35 - 42.
- [77] SAGYNTAY, Y. Informatics Olympiads in Kazakhstan: Team Selection and National. In: *Olympiads in Informatics*, 2022, vol. 16, pp. 135 - 143.
- [78] TSEDEVSUREN, D. Organization and Results of Mongolian National Online Olympiads in Informatics. In: *Olympiads in Informatics*, 2022, vol. 16, pp. 145 - 158.
- [79] GIORGIO AUDRITO, O. The Italian Job: Moving (Massively) Online a National Olympiad. In: *Olympiads in Informatics*, 2021, vol. 15, pp. 3 - 12.
- [80] HERNANDEZ GONZALES, F. The Cuban Olympiad in Informatics: A New Stage from the DMOJ Online Judge. In: *Olympiads in Informatics*, 2021, vol. 15, pp. 133 - 141.
- [81] GUTIERREZ, A. S. Argentine Olympiad in Informatics. In: *Olympiads in Informatics*, 2020, vol. 14, pp. 169 - 176.
- [82] *Moldova - results* [online]. Disponibil: <https://stats.ioinformatics.org/results/MDA>. [citat 16.03.2025].

- [83] ПЕРЕЛЬМАН, Я. *Живая математика. Занимательная алгебра*. Москва: Азбука, 2025.
- [84] ЧИСТЯКОВ, В. Д. *Старинные задачи по элементарной математике*. Минск: Высшая школа, 1978.
- [85] LOYD, S. *Mathematical Puzzles*. New York: Dover Publications, 1959.
- [86] DUDENEY, H. *The Caterbury Puzzles and other curios problems*. New York: Dover Publications, 1958.
- [87] GARDNER, M. *Mathematical Puzzles and diversions*. London: Bell and Sons, 1970.
- [88] ГАРДНЕР, М. *Математические досуги*. Москва Мир: Мир, 1972.
- [89] НОВИКОВ, Ф.А. *Дискретная математика для программистов*. Санкт Петербург : Питер, 2001.
- [90] ЛИПСКИЙ, В. *Комбинаторика для программистов*. Москва: Мир, 1988.
- [91] ЯЩЕНКО, В. В. *Введение в криптографию*. Санкт Петербург Питер: Питер, 2001.
- [92] МОЛДОВЯН, Н. *Практикум по криптосистемам с открытым ключом*. Санкт-Петербург: БХВ-Петербург, 2007.
- [93] CORLAT, A., **CORLAT, S.** *Calcul diferențial și Integral. Culegere de probleme*. Chișinău: Universitatea Academiei de Științe din Moldova, 2012.
- [94] ПЕРЕЛЬМАН, Я. *Занимательная физика*. Азбука, 2024.
- [95] BLOOM, B. *Taxonomy of Educational Objectives. The classification of Educational Goals*. London: Longmans, 1956.
- [96] BRUNER, J. *The process of Education*. Massachusetts: Harvard University Press, 1977.
- [97] O DECROLY, O., BOON, G. *Vers l'Ecole rénovée*. Paris: Soc. Cooperative Ane Etabi J: Lebegue & Cie, 36 Rue Neuve, Bruxelles, 1921.
- [98] FEIN, L. The Role of the University in Computers, Data Processing, and Related Fields. In: *Communications of the ACM*, 1960, vol. 2, nr. 9, pp. 7 - 14.
- [99] WIENER, N. *Cybernetics or control and communication in the animal and mashine*. Massachusetts: MIT Press, 1948, ediție 1965.
- [100] **CORLAT, S.**, IVANOV, L., BÂRSAN, V. *Informatica. Ghid de implementare a curriculumului modernizat pentru treapta liceală*. Chișinău: Cartier, 2010.
- [101] *The International Olympiad in Informatics Syllabus*. Version 2024. [online]. Disponibil: ioinformatics.org/files/ioi-syllabus-2024.pdf. [citât: 22 03 2025].

- [102] YOVCHEVA, B. *Spiral teaching programming to 10 – 11 year old pupils (based on the C++*. Sofia: Mathematics and mathematical education, 2007.
- [103] ПЕТРОВ, П. ШАЛАМАНОВ, М. *Състезателно програмиране за група С*. София: КЛМН, 2020.
- [104] КИРЮХИН, В. М. *Методика проведения и подготовки к участию в олимпиадах по Информатике*. Москва: Бином, 2011.
- [105] MAKISHEV, K. et al. *International Znamytkov Olympiad in Mathematics, Physics and Computer Science*. Almaty: Print - S, 2015.
- [106] CERCHEZ, E., ȘERBAN, M. *Programarea în limbajul C/C++ pentru liceu. Vol. I*. Iași: Polirom, 2005.
- [107] DIKS, K., LACKI, J. *Looking for a challenge?* Warsaw: University of Warsaw, 2012.
- [108] HALIM, S., HALIM, F. *Competitive Programming*. Singapore: Noble Education, 2021.
- [109] *Deutsches Spionage Museum* [online]. Disponibil: <https://www.deutsches-spionagemuseum.de/en/sammlung/the-cryptex>. [citat 02.04.2025].
- [110] COLLINS, J. *Numbers in minutes*. London: Quercus Editions Ltd., 2019.
- [111] GLENDINNING, P. *Maths in minutes. 200 key concepts, explained in an Instant*. London: Quercus Ltd., 2012.
- [112] БАПП, С. *Россыпи головоломок*. Москва: Мир, 1984.
- [113] *Școala virtuală a tânărului matematician* [online]. Disponibil: <https://www.math.md/school/distractiva/probleme/probleme.html>. [citat 26.03.2025].
- [114] *Laboratorul de Mate* [online]. Disponibil: <https://www.laboratoruldemate.com/category/matematica-distractiva/>. [citat 26.03.2025]
- [115] *Code* [online]. Disponibil: <https://code.org/>. [citat 26.03.2025].
- [116] *Math and Logic puzzles* [online]. Disponibil: <https://www.mathsisfun.com/puzzles/>. [citat 26.03.2025].
- [117] WORDERMAN, K. et al. *Programming for kids*. Moskova: Mann, Ivanov & Ferberr, 2016.
- [118] CRAIG, R. *Learn to Program with Minecraft: Transform Your World with the Power of Python*. San Francisco: No Starch press, 2015.
- [119] ПЕТРОВ, Ю. *История и философия науки. Математика, вычислительная техника, информатика*. Санкт-Петербург: БХВ-Петербург, 2005.
- [120] КАЗАКОВА, И. *История вычислительной техники : учеб. пособие*. Пенза: Изд-во ПГУ, 2011.

- [121] GARFINKEL, S. și GRUNSPAN, R. *THE COMPUTER BOOK: From the Abacus to Artificial Intelligence, 250 milestones in the history of Computing*. New York: Sterling, 2018.
- [122] CAMPBELL-KELLY, M. et al, *Computer A History of the Information Machine*. New York: Routledge, 2023.
- [123] CORTADA, J. *IBM The rise and Fall and Reinvention of a Global Icon*. Massachusetts: MIT Press, 2019.
- [124] *A Glossary of Computer Terms. 10-th edition*. Dorchester: Pearson Educated Limited, 2003.
- [125] TRIFAN, A. *Dicționar de Informatică*. București: Corint, 2010.
- [126] JODAL, E. *Dictionar de tehnica de calcul englez roman*. Cluj: Editura Albastră, 1998.
- [127] RĂILEANU, B. *Dicționar Englez - Român de termini tehnici și matematici*. București: Contemporary Literature Press, 2013.
- [128] eJOI. *First european junior olympiad in informatics* [online]. Disponibil: <https://ejoj.org/contest-technical-details/>. [citat 04.04.2025].
- [129] I *IOI Regulations 2024*. [online]. Disponibil: <https://www.ioi2024.eg/contest-rules>. [citat 04.04.2025].
- [130] DEITEL, H., DEITEL, P. *How to program in C / C++*. New Jersey: Prentice Hall, 2000.
- [131] ЙОВЧЕВА, Б. *Първи стъпки в програмирането на C/ C++*. Sofia: КЛМН, 2013.
- [132] ЙОВЧЕВА, Б. *Втори стъпки в програмирането на C/ C++*. Sofia, 2014.
- [133] CORLAT, S. *Programarea algoritmilor în C/C++*. [online]. Disponibil: <https://tekwill.online/courses/programarea-algoritmilor-in-c-c/>. [citat 04.04.2025].
- [134] BRAICOV, A. *Turbo-Pascal. Informatica. Culegere de probleme*. Chișinău: Prut, 2007.
- [135] ANASTASIU, D. P. *Informatica pentru gimnaziu, culegere de probleme (Pascal/C++)*. București, România: LS Info Mat, 2010.
- [136] CodeForces [online]. Disponibil: <https://codeforces.com/>. [citat 27.03.2025].
- [137] InfoArena [online]. Disponibil: <https://infoarena.ro/>. [citat 27.03.2025].
- [138] SKIENA, S. *The Algorithm Design Manual*. London: Springer, 2008.
- [139] Longest Good Array [online]. Available: <https://codeforces.com/problemset/problem/2008/C>. [citat 05.04.2025].
- [140] CORLAT, S. *Algoritmi și probleme de Geometrie Computațională*. Chișinău: Prut Internațional, 2009.

- [141] CHIRIAC, L. BOSTAN, M. *Elemente de Teoria Grafurilor*. Chișinău: Tipografia Centrală, 2022.
- [142] CORLAT S., GREMALSCHI, A. *Grafuri. Metodologia predării în cadrul instruirii de performanță la disciplinele Matematică și Informatică*. Chișinău: CEP Universitatea de Stat din Tiraspol, 2013.
- [143] CORLAT, S., CORLAT, A. *Grafuri. Noțiuni, algoritmi, implementări*. Chișinău: Biotehdesign, 2012.
- [144] SKIENA, S. și PEMMARAJU, S. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematic*. New York: Cambridge University Press, 2003.
- [145] CHIRIAC, L. *Competitive Geometry*. Chișinău: Prut Internațional, 2014.
- [146] CHIRIAC, L. *Metode Numerice*. Chișinău: Tipografia Centrală, 2014.
- [147] CORLAT, S. „BOI 2017,” [online]. Disponibil: <http://boi2017.utm.md>. [citat 07.04.2025].
- [148] CORLAT, S., GREMALSCHI, A. *Metodologia rezolvării problemelor de geometrie computațională*. Chișinău: Universitatea de Stat din Tiraspol, 2012.
- [149] КНУТ, Д. *Искусство программирования для ЭВМ. Сортировка и поиск*. Москва: Мир, 1978.
- [150] PRACSIU, D. *Culegere de probleme semnificative de Informatic*. Vaslui: Media Sind., 2015.
- [151] ПАПАДИМИТРИУ, Х. *Комбинаторная оптимизация. Алгоритмы и сложность*. Москва: Мир, 1985.
- [152] CORLAT, S. Algoritmi pe grafuri. Flux maxim în rețea. In: *Delta*, 2010, nr. 10, pp. 47-53.
- [153] BIGGS, N. L., LLOYD, E., WILSON, R. J. *Graph Theory, 1736 - 1936*. Oxford: Clarendon Press, 1976.
- [154] CADAR, C., STROE, M. *Culegere de probleme și programe Pascal*. București: Petrion, 1996.
- [155] IȘVAC, C., PRUNĂ, M. *Bazele Informaticii. Grafuri și Combinatorică*. București: Petrion, 1990.
- [156] BELLMAN, R. *Dynamic Programming, Princeton*. New Jersey: Princeton University Press, 1957.
- [157] BELLMAN, R. *Adaptive Control Processes: A Guided Tour*. Princeton : Princeton University Press, 1961.

- [158] BELLMAN, R., KABALLA, R. *Dynamic Programming and Modern Control Theory*. Academic Press: Academic Press, 1965.
- [159] DIJKSTRA, E. W. A note on two problems in connexion with graphs. In: *Numerische Mathematik*, 1959.
- [160] DIJKSTRA, E. W. *A Discipline of Programming*. Prentice-Hall, 1976.
- [161] SORIN, T. *Concursuri de programare. Probleme și soluții*. București: Teora, 1997.
- [162] SKIENA, S. *Programming Challenges*. New York: Springer, 2003.
- [163] CORLAT, S., DANIEL, G. An Analysis of the problems from Balkan Olympiad in Informatics. 2014. In: *Învățământul de performanță la disciplinele din ariile curriculare Științe Exacte și Științe ale Naturii. Obiective. Strategii. Perspective*, Chișinău, 2014.
- [164] CORLAT, S., ȚARIGRADSCHI, M. International Olympiad in Informatics (IOI) 2014. Tasks and Solutions. In: *Învățământul de performanță la disciplinele din ariile curriculare Științe Exacte și Științe ale Naturii. Obiective. Strategii. Perspective*, Chișinău, 2014.
- [165] ESANU, M., CORLAT, S. A comparative study of maximum matching algorithms. In: *Acta Et Commentationes*, 2024.
- [166] CORLAT, S. Aplicații ale Geometriei computaționale în rezolvarea problemelor de Informatică. In: *Delta*, 2006, nr. 2, pp. 13-18.
- [167] CORLAT, S., BOSTAN, M. Algoritmi de colorare a grafurilor. Analiza eficienței. In: *Învățământul de performanță la disciplinele din ariile curriculare Științe Exacte și Științe ale Naturii. Obiective. Strategii. Perspective*, Chișinău, 2014.
- [168] HORVATH, G. IOI Syllabus. [online]. Disponibil: <https://ioinformatics.org/files/ioi-syllabus-2024.pdf>. [citat 27.03.2025].
- [169] GLOBA, A., CORLAT, S. Didactic aspects regarding to creating test sets for competition problems. *Acta et commentationes (Științe ale Educației)*, 2019, vol. 4, nr. 18.
- [170] AtCoder [online]. Disponibil: <https://atcoder.jp/>. [citat 27.03.2025].
- [171] Punct Campion. [online]. Diponibil: <http://campion.edu.ro/index.php>. [citat 27.03.2025].
- [172] CORLAT, S. Integrarea portofoliilor de predare și învățare. Metaportofoliul. In: *Didactica Pro*, 2010, pp. 8-12.
- [173] CircuitVerse [online]. Disponibil: <https://circuitverse.org/simulator>. [citat 09.04.2025].
- [174] Code [online]. Disponibil: <https://code.org/>. [citat 01.04.2025].
- [175] International Advanced Tournament in Informatics. [online]. Disponibil: <https://iati-shu.org/>. [citat 27.03.2025].

- [176] GHICOV, A. *Pedagogia aplicativă a performanței*, Chișinău: Pontos, 2012.
- [177] CHAUVE, P. et al. *Learning and teaching in the communication society*. Strasbourg: Council of Europe, 2003.
- [178] *Concursul de Fizică. In memoriam Mihai Marinciuc*. [online]. Disponibil: https://fizica.utm.md/data/concurs_marinciuc.php. [citat: 27.03.2025].
- [179] RENZULLI, J. The Three-Ring Conception of Giftedness: A Developmental Model For Promoting Creative Pro-ductivity. In: *Reflections on gifted education*. 2016, Waco, Prufrock Press.
- [180] CALLO, T. Învățarea racordată – factor al excelenței cognitive a elevului merituos. In: *Didactica Pro*, 2021. nr. 4-5 (128 - 129).
- [181] CIOBAN, M., LUPU, I., SALI, L. Educația de performanță: Contexte. Obiective. strategii. In: *Revista de Știință, Inovare, Cultură și Artă „Akademos”*, 2014, nr. 4(35).
- [182] SALI, L., TELEUCA, M. Design-ul domeniilor de dezvoltare a elevilor capabili de performanțe înalte la matematică. In: *Mathematics and Information Technologies: Research and Education*, 2023, Chișinău.
- [183] GARDNER, H. *Multiple Intelligences: The Theory in Practice*. Paidos Iberica Ediciones, 1995.
- [184] BANDURA, A. Self-efficacy: Toward a Unifying Theory of Behavioral Change. In: *Psychological Review*, 1977, vol. 84, nr. 2.
- [185] GARDNER, H. *Inteligente multiple. Noi perspective*. București: Curtea Veche Publishing, 2022.
- [186] SCHUNK, D. Motivation and social cognitive theory. In: *Contemporary Educational Psychology*, 2020, vol. 60.
- [187] ANDERSON, J. *Cognitive Psychology and its Implications*. New York: Worth Publishers, 2010.
- [188] DECI, E., RYAN, R. *Intrinsic Motivation and Self-Determination in Human Behavior*. New York: Plenum Press, 1985.
- [189] PEKRUN, R. *The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practic*. In: *Educational Psychology Review*, 2006, nr. 18(4):315-341.
- [190] McAULEY, E. et al. Growth trajectories of exercise self-efficacy in older adults: Influence of measures and initial status. In: *Health Psychology*, 2011, nr. 30(1), 75–83.

- [191] ZIMMERMAN, B. Becoming a Self-Regulated Learner: an Overview. In: *Theory Into Practice* , 2002, vol. 41, nr. 2. doi: 10.1207/s15430421tip4102_2
- [192] WEINTROP, D. et al. Defining Computational Thinking for Mathematics and Science Classrooms. In: *Journal of science education and technology*, 2016, vol. 25, nr. 2.
- [193] PAPERT, S. *Children, Computers and Powerful Ideas*, New York, New York: Basic Books, 1980.
- [194] WING, J. Viewpoint: Computational thinking. In: *Communications of the ACM*, 2006, vol. 49, nr. 3(2006), pp. 33-35.
- [195] FACIONE, P. *Critical Thinking: What It Is and Why It Counts*. 2011. [online]. <https://www.researchgate.net/publication/251303244>
- [196] GUILFORD, J. *The nature of Human Intelligence*, New York: MckGraw-Hill Book Company, 1967.
- [197] *Regulamentului privind activitățile de susținere a elevilor capabili de performanțe înalte în învățământul general*. HG nr. 1002, din 13-12-2023. [online]. Disponibil: https://www.legis.md/cautare/getResults?doc_id=140564&lang=ro. [citat: 05.04.2025].
- [198] DARWIN, C. *The ORigin Of Species*, Londra: John Murray, 1859.
- [199] PIAGET, J., INHELDER, B. *The psychology of the Child*. New York: Basic Books, 1969.
- [200] GINSBURG, H., OPPER, S. *Piaget's Theory of Intellectual Development*. International Psychotherapy Institute E-Books, 2016.
- [201] SIEGLER, R. *Children's thinking. 2nd ed*. New Jersey: Prentice Hall, 1991.
- [202] *QS World University Rankings 2025: Top global universities 2025*. [online]. Disponibil: <https://www.topuniversities.com/>. [citat 01 04 2025].

ANEXE

Anexa 1. Soluția informatică a problemei Tale (Poveste), propusă la Olimpiada Balcanică de Informatică, 2017

```
#include <bits/stdc++.h>
#include <cmath>
#define fl double
using namespace std;
const fl pi = acos(-1.0);
const fl eps = 0.000000001;
int n;
int prv(int i){return i==1?n:i-1;}
int nxt(int i){return i==n?1:i+1;}
int i,j,st,dr,gasit,cnt_st,cnt_dr;
fl X, Y, L, u, sol = 0, x[1010], y[1010], aria(fl,fl,fl,fl,fl,fl),
unghi(fl,fl,fl,fl,fl,fl), dist(fl,fl,fl,fl);
void vizibil(fl,fl,fl,fl,fl&,fl&);

int cmp(fl a, fl b)
{
    if (a + eps < b) return -1;
    if (b + eps < a) return 1;
    return 0;
}

int main()
{
    cin >> X >> Y >> L >> n;
    for(i = 1; i <= n; i++)
    { cin >> x[i] >> y[i]; x[i] -=X; y[i] -=Y;}
    for(i = 1; i <= n; i++)
    {
        fl ad = aria(0,0,x[i],y[i],x[prv(i)],y[prv(i)]);
        fl as = aria(0,0,x[nxt(i)],y[nxt(i)],x[i],y[i]);
        if(cmp(ad, 0) < 0 &&cmp(as, 0) >= 0){st = i; gasit++;}
        if(cmp(as, 0) < 0 && cmp(ad, 0) >= 0){dr = i; gasit++;}
        if(gasit == 2) break;
    }

    fl un = 2 * pi;
    for(i = dr, j = nxt(dr); i != st; i = nxt(i),j = nxt(j))
    {
        fl ar, un;
        vizibil(x[j],y[j],x[i],y[i],ar, un);
        sol +=ar;
        un -=un;
    }
    sol += Un*L*L/2.0;
```

```

fl RS, RD; int ps, pd, ss, sd;
RS = L - dist(0,0,x[st],y[st]); ps = prv(st);ss = nxt(st);
while(cmp(RS, 0) >0)
{
    cnt_st++;
    if(st == dr) return 0;
    fl RC = dist(x[st],y[st],x[ss],y[ss]);
    fl UC = unghi(x[ps],y[ps],x[st],y[st],x[ss],y[ss]);
    if(cnt_st == 1)
        UC = unghi(0.0,0.0,x[st],y[st],x[ss],y[ss]);
    if(cmp(RS, RC) < 0)
    { sol +=UC * RS * RS / 2.0; break; }
    sol +=UC * RS * RS / 2.0;
    RS -=RC;
    ps = nxt(ps); st = nxt(st); ss = nxt(ss);
}
RD = L - dist(0,0,x[dr],y[dr]); pd = nxt(dr);sd = prv(dr);
while(cmp(RD, 0) > 0)
{
    cnt_dr++;
    if(st == dr) return 0;
    fl RC = dist(x[dr],y[dr],x[sd],y[sd]);
    fl UC = unghi(x[pd],y[pd],x[dr],y[dr],x[sd],y[sd]);
    if(cnt_dr == 1) UC = unghi(0,0,x[dr],y[dr],x[sd],y[sd]);
    if(cmp(RD, RC) < 0)
    {
        sol +=UC * RD * RD / 2.0; break;
    }
    sol +=UC * RD * RD / 2.0;
    RD -=RC;
    pd = prv(pd); dr = prv(dr); sd = prv(sd);
}
if(ss == dr)
{
    fl DC = dist(x[st],y[st],x[nxt(st)],y[nxt(st)]);
    if(cmp(RS+RD, DC) > 0)
        sol-=RS*RS*acos((RS*RS+DC*DC-RD*RD)/(2*RS*DC))/2.0+
        RD*RD*acos((RD*RD+DC*DC-RS*RS)/(2*RD*DC))/2.0-
        sqrt((DC+RS+RD)*(-DC+RS+RD)*(DC-RS+RD)*
            DC+RS-RD))/4.0;
}
return 0;
}
fl aria(fl a1,fl b1,fl a2,fl b2,fl a3,fl b3)
{
    return a1*b2+a2*b3+a3*b1-b1*a2-b2*a3-b3*a1;
}
fl dist(fl a1,fl b1,fl a2,fl b2)

```

```

{
    return sqrt((a2-a1)*(a2-a1)+(b2-b1)*(b2-b1));
}
fl unghi(fl a1,fl b1,fl a2,fl b2,fl a3,fl b3)
{
    fl a, b, c; a = dist(a1,b1,a2,b2); b = dist(a2,b2,a3,b3); c =
    dist(a3,b3,a1,b1);
    return pi - acos ((a * a + b * b - c * c)/(2 * a * b));
}
void vizibil(fl a1,fl b1,fl a2,fl b2,fl &ar, fl &un)
{
    fl a, b, c, h; c = dist(a1,b1,a2,b2);
    if(c < eps) {ar = eps; un = eps; return;}
    if(cmp(fabs(a1 * b2-a2 * b1), L * dist(a1, b1, a2, b2))>=0)
        {ar = eps; un = eps; return;}
    a = dist(0,0,a1,b1); b = dist(0,0,a2,b2);
    if(cmp((a-L)*(b-L),0)<0)
    {fl ar1,ar2,un1,un2; a = (a1 + a2) / 2.0;
        b = (b1 + b2) / 2.0;
        vizibil(a1,b1,a,b,ar1,un1); vizibil(a,b,a2,b2,ar2,un2);
        ar = ar1 + ar2; un = un1 + un2; return;
    }
    if(cmp(a, L) <= 0 && cmp(b, L) <= 0)
    { ar = fabs(a1 * b2 - b1 * a2) / 2.0;
        un = acos((a * a + b * b - c * c) / (2.0 * a * b));return;
    }
    if(cmp(a * a + c * c, b * b) < 0 || cmp(b * b + c * c, a * a) <
    0){ar = eps; un = eps; return;}
    a = b = L; h = fabs(a1 * b2 - a2 * b1) / dist(a1, b1, a2, b2);
    c = 2.0 * sqrt(L * L - h * h); ar = c * h / 2.0; un = acos((a * a
    + b * b - c * c)/(2 * a * b));
}

```

Anexa 2. Chestionarul pentru analiza activității de performanță după absolvirea liceului

Chestionar Implicare în mișcarea olimpică după absolvirea liceului

B *I* U  

Form description

Universitatea: care este universitatea absolvită (sau la care înveți) *

Short answer text

Competițiile studentești de programare: participi singur, sau în componența unei echipe universitare? *

☐ Da

☐ Nu

Participi (ai participat) în calitate de membru sau Voluntar în activitatea Consiliului Național Olimpic la Informatică? *

☐ Da

☐ Nu

Participi (ai participat) în calitate de membru sau Voluntar în organizarea concursurilor la Informatică la nivel național? *

☐ Da

☐ Nu

Participi (ai participat) în calitate de membru sau Voluntar în organizarea concursurilor la Informatică la nivel internațional? *

☐ Da

☐ Nu

Propui / ai propus probleme pentru concursuri internaționale de Informatică? *

☐ Da

☐ Nu

Intenționezi să te implici în continuare activ în activitățile olimpice la Informatică *

☐ Da, ca membru al Consiliului Național Olimpic

☐ Da, ca autor de probleme

☐ Da, ca lider / Deputy leader al echipelor

☐ Da, ca mentor

☐ Nu

DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII

Subsemnatul, declar pe răspundere personală că materialele prezentate în teza de doctorat sunt rezultatul propriilor cercetări și realizări științifice. Conștientizez că, în caz contrar, urmează să suport consecințele în conformitate cu legislația în vigoare.

Numele, prenumele Sergiu Corlat

Semnătura



Data

26.05.2025

CURRICULUM VITAE A CANDIDATULUI



Europass Curriculum Vitae



Informații personale

Nume **Corlat Sergiu**
Adresă 2/4, str. Ceucari, ap. 50, MD2020, Chișinău, Moldova
Telephone(s) (+373) 6088 9088 Mobile: (+373) 68818825
E-mail sergiu.corlat@gmail.com sergiu.corlat@usm.md

Data nașterii Ianuarie, 23, 1967

Gen Male

Experiență de muncă

Date 2021 - prezent
Instituție / funcție Universitatea de Stat din Moldova, Prorector
Responsabilități de bază Implementarea Strategiei de Dezvoltare Digitală a Universității
Nume și adresă angajator Str. A. Mateevici, 60, MD-2009, Chișinău, Republica Moldova
Domeniu de activitate Educație

Date 2018 - 2021
Instituție / funcție Universitatea Tehnică a Moldovei, lector universitar
Responsabilități de bază Cursuri: Programarea calculatoarelor, Algoritmi și structuri de date
Nume și adresă angajator Bd. Ștefan cel Mare, 168, MD-2004, Chișinău, Republica Moldova
Domeniu de activitate Educație

Date 2016 - 2018
Instituție / funcție Fundația Y4Innovation. Expert în educația pentru TI.

Responsabilități de bază Cursuri: Programare de performanță, Pregătire pentru examenul național la Informatică.

Nume și adresă angajator Fundația Y4Innovation, 170, Columna str., Chișinău, Moldova

Domeniu de activitate Educație

Date 1992 – 1995, 1997 – 2016

Instituție / funcție Universitatea de Stat din Moldova . Lector universitar, FMI, FJSC

Responsabilități de bază Cursuri: ASD, Geometrie computațională, e-comunicare, Desktop publishing, Web design

Nume și adresă angajator Universitatea de Stat din Moldova , 60, Mateevici str. MD2009, Chișinău, Moldova

Domeniu de activitate Educație

Date 2006 - 2010

Instituție / funcție Centrul Tehnologii Informaționale și de Comunicații în Educație. Vice director.

Responsabilități de bază Managementul general al Instituției. Implementarea TIC în Educație.

Nume și adresă angajator Centrul Tehnologii Informaționale și de Comunicații în Educație. 1, A. Russo str., MD2068, Chișinău, Moldova

Domeniu de activitate Educație

Date 2006 - 2019

Instituție / funcție Universitatea de Stat Tiraspol
Lector universitar(2007 - 2018), Șef al Centrului Tehnologii Informaționale UST (2016 - 2018)

Responsabilități de bază Cursuri: Algoritmi și structuri de date, Geometrie Computațională, programare C++

Nume și adresă angajator Universitatea de Stat Tiraspol. 5, str. Iabloikin, MD2069, Chisinau, Moldova

Domeniu de activitate Educație

Date 2000 - prezent

Instituție / funcție Liceul Teoretic „Orizont” . Profesor de Informatică.

Responsabilități de bază Cursuri: Informatică, Algoritmi și structuri de date, Geometrie Computațională, programare C++

Nume și adresă angajator Liceul Teoretic „Orizont” . 22, str. Dimo, Durlești, Chișinău, Moldova

Domeniu de activitate Educație

Date	1995 - 1996
Instituție / funcție	Editura „Prut International”. Șef sector pre Press.
Responsabilități de bază	Elaborare design ediții tipografice
Nume și adresă angajator	Editura „Prut International”, 59, I. Creanga str., Chișinău, Moldova
Domeniu de activitate	Edituri, poligrafie

Educație

Date	01.09.2023 – 02.12.2023		
Titlu / calificare	Cursuri de calificare. Domeniu: matematica / 0541.2		
Subdomeniu / abilități	Matematica Aplicată / Robotică și Mecatronică (educator pentru Dorne)		
Instituția care a oferit formarea	Universitatea de Stat din Moldova		
Date	06/26 – 07 / 07 2016,	06/11 – 06 / 17 2017,	06/11 – 06 / 16 2018
Titlu / calificare	dezvoltarea resurselor educaționale digitale pentru rețele		
Subdomeniu / abilități	naționale ale Centrelor de formare continuă a cadrelor didactice		
Instituția care a oferit formarea	Universitatea Kingston, Marea Britanie , Universitatea Sofia Antipolis, Nice, Franța, Universitatea din Girona, Spania		
Date	11/23 – 12 / 03 2015		
Titlu / calificare	Programare vizuală, Robotică, Elaborare resurse educaționale		
Subdomeniu / abilități	digitale		
Instituția care a oferit formarea	Oficiul Special Guvernamental pentru Educație, Coreea		
Date	2012 - 2014		
Titlu / calificare	Master în Științe Exacte.		
Subdomeniu / abilități	Informatica Aplicată.		
Instituția care a oferit formarea	Universitatea Academiei de Științe, Moldova		
Date	05/22 – 05/30 2012		
Titlu / calificare	Modernizarea sistemului educațional din Republica Moldova		
Subdomeniu / abilități			
Instituția care a oferit formarea	Istropolitan Nova Academy, Bratislava, Slovacia		
Date	10/15 – 10/22 2010	01/23 – 01/29 2010	11/15 – 11/22 2009

Titlu / calificare	Metodologii e-Learning
Subdomeniu / abilități	
Instituția care a oferit formarea	KTU, Kharkiv University of Technology, Ukraine, KTH, Royal Institute of Technology, Stockholm, Sweden, Kaunas University of Technology, Lithuania
Date	09/12 – 09/19 2008
Titlu / calificare	Abordări sistemice pentru strategia integrării TIC în educație
Subdomeniu / abilități	
Instituția care a oferit formarea	E-Governance Academy, Tallinn, Estonia
Date	06/30 – 07/04 2008
Titlu / calificare	Universitatea de Vară pentru Democrație, Școala de Studii Politice a Consiliului Europei
Subdomeniu / abilități	
Instituția care a oferit formarea	Consiliului Europei, Strasbourg, France
Date	01/20 – 02/14 2005
Titlu / calificare	ICT în Jurnalism și Științele Comunicării
Subdomeniu / abilități	
Instituția care a oferit formarea	Universitatea Columbia, Missouri, USA
Date	08/14 – 08/20 2004
Titlu / calificare	Școala de vară „Dezvoltarea Resurselor Educaționale Software”
Subdomeniu / abilități	
Instituția care a oferit formarea	„Siveco” SA, Drobeta Turnu Severin, Romania
Date	03/03 – 03/15 1997, 02/10 – 03/05 1998
Titlu / calificare	Noi Tehnologii Informaționale în Jurnalism
Subdomeniu / abilități	
Instituția care a oferit formarea	Complutense University, Madrid, Spain
Date	1984 - 1992 (1985 – 1988 întrerupere – serviciul militar, flota militară)
Titlu / calificare	Matematician
Subdomeniu / abilități	Matematica Aplicată
Instituția care a oferit formarea	Facultatea de Matematică și Cibernetică, Universitatea de Stat din Moldova

Competențe personale

Limba maternă	Română				
Limbi străine					
Auto-evaluare	Înțelegere		Vorbire		Scriere
<i>European level (*)</i>	Ascultare	Citire	Discuți	Ascultare	
Rusa	C2	C2	C2	C2	C2
Engleza	B2	B2	B2	B2	B2

Abilități și competențe sociale Comunicabil, dispus să asculte punctele de vedere ale altora, să analizeze diferite puncte de vedere și să aleagă soluția optimă.

- Competențe organizaționale**
- Proiectul Băncii Mondiale pentru Învățământul Superior în Moldova. Director al subproiectului Universitatea de Stat din Moldova: "IT-HUB". (2022 - 2025)
 - Conducător al proiectului "Biblioteca Digitală USM", program KUSANONE, Ambasada Japoniei în Moldova, 2023 - 2024
 - Expert al fundației Soros Moldova pentru elaborarea studiului de politici educaționale "The learning process in the conditions of educational actors' isolation", 2020
 - Formator național în cadrul proiectului "Competențe digitale pentru ocuparea forței de muncă în economia modernă", PRO Didactica, 2019 - 2020
 - Expert național în cadrul proiectului "Dezvoltarea și modernizarea curriculară", Modul: Informatica. Ministerul Educației, Culturii și Cercetării al Republicii Moldova, 2019
 - Expert național în cadrul proiectului "Dezvoltarea și modernizarea curriculară", Modul: Educația Digitală. Coordonator al grupului. Ministerul Educației, Culturii și Cercetării al Republicii Moldova, 2018
 - Conducător al echipei Republicii Moldova la Olimpiada Europeană de Informatică pentru fete (Switzerland, 2021, online)
 - Organizarea competițiilor de programare între colegii, Zona Centru - Sud (2008 – 2020, 2024), Președinte al comitetului științific.
 - Conducător al echipei Republicii Moldova la Olimpiada Internațională de Informatică (2014 – Taiwan, Taipei, 2015 – Kazahstan, Almaty, 2016 – Kazan, Rusia, 2017 – Teheran, Iran, 2018 – Tsukuba, Japan), 2021 – Singapore (on-line ediția), 2022 – Yogikarta, Indonezia, 2023 – Szeged, Hungary, 2024 – Alexandria, Egipt.

- Conducător al echipei Republicii Moldova la Olimpiada Balcanică de Informatică (2003 - Romania, 2004 – Bulgaria, 2006 – Cyprus, 2009 – Bulgaria, 2016 – Macedonia (JBOI), 2018 – Romania (JBOI)). 2023 - Slovenia
- Conducător al echipei Republicii Moldova la Olimpiada de Informatică a Europei Centrale (2009, 2016 – Romania).
- Membru al Comitetului de Organizare al EJOI 2024, BOI 2007, 2017, Chișinău, Moldova
- Coordonator al echipei Republicii Moldova la Concursul Internațional de Proiecte informatice „Infomatrix” (2004 – „Graph magic” proiect, 2005 – „G-derivator proiect”, 2011 – ”Rescue Robot ”).
- Coordonator al proiectului ”FlyPy”. INTEL ISEF, 2014, California, USA.