



Ministerul Educației și Cercetării al Republicii Moldova
Centrul de Excelență în Informatică și Tehnologii Informaționale

"Aprob"

Directorul Centrului de Excelență în Informatică și
Tehnologii Informaționale

_____ Andrei Ciobanu

" ____ " _____ 2025

Curriculumul modular

F.03.O.013 Programarea calculatorului

F.02.O.013 Programarea calculatorului

Specialitatea: **0611.2 Rețele de calculatoare**

Aprobat:

La ședința Consiliului metodic-științific din 20.06.2025 al Centrului de Excelență în Informatică și Tehnologii Informaționale, proces verbal Nr. 7

Director adjunct _____ Obadă Liuba

La ședința Catedrei de Informatică II din 14.05.2025 al Centrului de Excelență în Informatică și Tehnologii Informaționale, proces verbal Nr. 10

Șef catedra _____ Gribineț Luminița

Coordonat cu:

Centrul de Excelență în Informatică și Tehnologii Informaționale

Centrul de Excelență în Energetică și Electronică

Colegiul Politehnic din Bălți

Colegiul Universității Tehnice a Moldovei

Autori:

Gîncu Silviu, doctor în pedagogie, grad didactic superior, Centrul de Excelență în Economie și Finanțe

Șarapanovscaia Irina, grad didactic întâi, Centrul de Excelență în Informatică și Tehnologii Informaționale

Siminițchi Ghorghie, grad didactic doi, Colegiul Politehnic din Bălți

Recenzenți

Sudacevschi Viorica, șefa departamentului Informatică și Ingineria Sistemelor, UTM

Rogojinaru Tatiana, manager serviciul customer, StarLab SRL

Cuprins

I. Preliminarii	4
II. Motivația, utilitatea unității de curs pentru dezvoltarea profesională	4
III. Competențele profesionale și rezultatele învățării	5
IV. Administrarea unității de curs	5
V. Repartizarea orientativă a orelor pe unități de învățare	6
VI. Unitățile de învățare	7
VII. Studiu individual ghidat de profesor	12
VIII. Lucrările practice recomandate	12
IX. Sugestii metodologice	12
X. Sugestii de evaluare	15
XI. Resurse necesare pentru atingerea rezultatelor învățării	17
XII. Resursele didactice recomandate elevilor	18

I. Preliminarii

Cursul „Programarea calculatorului” face parte din domeniul învățământului profesional tehnic post-secundar și pregătește specialiști pentru calificarea de *tehnician/tehniciană suport tehnic al rețelelor*. Această calificare vizează formarea competențelor necesare pentru dezvoltarea, mentenanța și optimizarea produselor software utilizate în diverse domenii economice și tehnologice. Absolvenții vor putea gestiona structuri dinamice de date, implementa tehnici de programare și analiza algoritmi în scopul dezvoltării unor aplicații performante și eficiente.

Funcțiile Curriculumului. Funcțiile de bază ale Curriculumului sunt:

- act normativ al procesului de predare, învățare, evaluare și certificare în contextul unei pedagogii axate pe competențe;
- reper pentru proiectarea didactică și desfășurarea procesului educațional din perspectiva unei pedagogii axate pe competențe;
- componentă de bază pentru elaborarea strategiei de evaluare și certificare;
- orientare a procesului educațional spre formare de competențe la elevi;
- componentă fundamentală pentru elaborarea manualelor tipărite, manualelor electronice, ghidurilor metodologice, testelor de evaluare.

Beneficiarii Curriculumului. Curriculumul este destinat:

- profesorilor din instituțiile de învățământ profesional tehnic post secundar;
- autorilor de manuale și ghiduri metodologice;
- elevilor care își fac studiile la specialitatea în cauză;
- membrilor comisiilor pentru examenele de calificare;
- membrilor comisiilor de identificare, evaluare și recunoaștere a rezultatelor învățării, dobândite în contexte non-formale și informale.

Scopul principal al modului este de a forma specialiști capabili să utilizeze tipurile dinamice de date și tehnicile avansate de programare în proiectarea și implementarea soluțiilor informatice. De asemenea, modulul urmărește dezvoltarea gândirii algoritmice, a abilităților de analiză și optimizare a aplicațiilor software, precum și familiarizarea cu standardele și normele din domeniul dreptului informatic.

Prin parcurgerea acestui modul, elevii vor dobândi competențele necesare pentru a contribui la proiectarea și întreținerea produselor software, pregătindu-se astfel pentru integrarea pe piața muncii sau pentru continuarea studiilor în domeniul IT.

II. Motivația, utilitatea unității de curs pentru dezvoltarea profesională

Unitatea de curs „Programarea calculatorului” joacă un rol esențial în formarea competențelor necesare pentru calificarea de *tehnician/tehniciană suport tehnic al rețelelor* și pentru dezvoltarea unei cariere de succes în domeniul tehnologiilor informaționale. Aceasta oferă elevilor cunoștințele și abilitățile fundamentale pentru utilizarea structurilor dinamice de date, aplicarea tehnicilor de programare și gestionarea resurselor informatice, competențe esențiale în dezvoltarea de produse software eficiente și scalabile.

Prin studiul acestei unități, elevii vor dobândi:

- *Cunoștințe fundamentale* despre structuri de date dinamice, algoritmi de bază și tehnici de optimizare a programelor informatice.
- *Abilități cognitive și practice* necesare pentru dezvoltarea, mentenanța și testarea aplicațiilor de consolă.
- *Capacitatea de analiză și rezolvare a problemelor* prin implementarea tehnicilor de programare adecvate fiecărei situații.
- *Dezvoltarea gândirii algoritmice* și utilizarea corectă a principiilor de proiectare software.

Aceste competențe vor permite elevilor să abordeze sarcini de lucru complexe, să înțeleagă și să implementeze soluții eficiente și să se adapteze cerințelor pieței muncii din domeniul IT.

Astfel, această unitate de curs nu doar că asigură condițiile necesare pentru studiul disciplinelor avansate, dar și contribuie la formarea unui fundament solid pentru integrarea elevilor pe piața muncii și dezvoltarea unei cariere profesionale de succes în domeniul IT.

III. Competențele profesionale și rezultatele învățării

În cadrul modului vor fi formate și dezvoltate următoarele competențe profesionale:

CP9. Utilizarea software pentru configurarea echipamentelor de rețea Utilizarea algoritmilor pr grafuri la rezolvarea problemelor complexe.

CP10. Aplicarea metodelor de testare a funcționalității rețelei.

Absolventul programului la atribuirea calificării poate:

RÎ9. Configura echipamentele active de rețea prin aplicații software dedicate, implementând setările necesare pentru funcționarea optimă a rețelei conform cerințelor specificate, normele de securitate și compatibilitate.

RÎ10. Aplica metode de testare pentru a evalua funcționalitatea rețelei, identificând și diagnosticând problemele posibile, oferind soluții de remediere în cel mai scurt timp.

După finalizarea acestui modul, elevii vor fi capabili să:

RÎ1. Implementeze structuri dinamice de date în aplicații de consolă, gestionând eficient memoria și resursele, conform specificațiilor tehnice.

RÎ2. Aplice tehnici avansate de programare pentru rezolvarea problemelor algoritmice, asigurând corectitudinea, optimizarea și structura clară a codului.

RÎ3. Optimizeze performanța algoritmilor implementați, folosind concepte avansate de complexitate algoritmică pentru îmbunătățirea soluțiilor software.

RÎ4. Dezvolte algoritmi eficienți pentru reprezentarea și procesarea grafurilor în aplicații de consolă, minimizând consumul de resurse prin metode adecvate de stocare și prelucrare a datelor.

IV. Administrarea unității de curs

Învățământ clasic (4 ani)

Sem estrul	Numărul de ore				Forma de evaluare	Numărul de credite
	Total ore	Contact direct		Studiu individual		
		Teorie	Laborator			

III	90	30	30	30	Examen	3
-----	----	----	----	----	--------	---

Învățământ clasic (2 ani)

Semestrul	Numărul de ore				Forma de evaluare	Numărul de credite
	Total ore	Contact direct		Studiu individual		
		Teorie	Laborator			
II	90	30	30	30	Examen	3

V. Repartizarea orientativă a orelor pe unități de învățare

Învățământ clasic (4 ani)

Nr. crt.	Unități de învățare	Numărul de ore			
		Total ore	Contact direct		Studiu individual
			Teorie	Laborator	
1.	Structuri dinamice de date	36	12	12	12
2.	Tehnici de programare	36	12	12	12
3.	Elemente de teoria grafurilor	18	6	6	6
	Total	90	30	30	30

Învățământ clasic (2 ani)

Nr. crt.	Unități de învățare	Numărul de ore			
		Total ore	Contact direct		Studiu individual
			Teorie	Laborator	
1.	Structuri dinamice de date	36	12	12	12
2.	Tehnici de programare	36	12	12	12
3.	Elemente de teoria grafurilor	18	6	6	6
	Total	90	30	30	30

VI. Unitățile de învățare

Unități de competență	Unități de conținut	Abilități
1. Structuri dinamice de date		
	Structuri dinamice de date. Tipuri dinamice de date: - listă simplu înlănțuite; - liste dublu înlănțuite; - liste circulare. Operații specifice tipurilor dinamice de date: - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor.	A1. Declararea unei structuri dinamice de date. A2. Alocarea dinamică a memoriei unei variabile dinamice. A3. Eliberarea memoriei dinamice alocate unei variabile dinamice. A4. Crearea unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A5. Afișarea datelor unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A6. Căutarea datelor unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A7. Permutarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A8. Eliminarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A9. Adăugarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A10. Interclasarea structurilor dinamice de date. A11. Ordonarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse. A12. Distrugerea unei structuri dinamice de date de tip listă. A13. Elaborarea algoritmilor pentru tipuri de date de tip listă. A14. Translarea algoritmilor pentru tipuri de date de tip listă în limbajul de programare. A15. Implementarea algoritmilor pentru tipuri de date de tip listă în limbajul de programare.

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități
	Structuri dinamice de date prestabilite: - stivă; - coadă.	A16. Crearea unei structuri dinamice de date de tip stivă, coadă. A17. Afișarea datelor unei structuri dinamice de date de tip stivă, coadă. A18. Căutarea datelor unei structuri dinamice de date de tip stivă, coadă.

Unități de competență	Unități de conținut	Abilități
	Operații specifice tipurilor dinamice de date: - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor.	A19. Eliminarea datelor în cadrul unei structuri dinamice de date de tip stivă, coadă. A20. Adăugarea datelor în cadrul unei structuri dinamice de date de tip stivă, coadă. A21. Distrugerea unei structuri dinamice de date de tip stivă, coadă. A22. Elaborarea algoritmilor pentru tipuri de date de tip stivă, coadă. A23. Translarea algoritmilor pentru tipuri de date de tip stivă, coadă în limbajul de programare. A24. Implementarea algoritmilor pentru tipuri de date de tip stivă, coadă în limbajul de programare.
	Arbori binari. Operații specifice tipului de date arbori binari: - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor. Arbori binari de căutare. Operații specifice: - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere.	A25. Crearea unei structuri dinamice de date de tip arbore binar. A26. Afișarea datelor unei structuri dinamice de date de tip arbore binar. A27. Căutarea datelor unei structuri dinamice de date de tip arbore binar. A28. Eliminarea datelor în cadrul unei structuri dinamice de date de tip arbore binar. A29. Adăugarea datelor în cadrul unei structuri dinamice de date de tip arbore binar. A30. Distrugerea unei structuri dinamice de date de tip arbore binar. A31. Parcurgerea arborilor binari prin intermediul metodelor: în lățime, în lungime, în adâncime. A32. Elaborarea algoritmilor pentru tipuri de date de tip arbore binar. A33. Translarea algoritmilor pentru tipuri de date de tip arbore binar în limbajul de programare. A34. Implementarea algoritmilor pentru tipuri de date de tip arbore binar în limbajul de programare.

Unități de competență	Unități de conținut	Abilități
2. Tehnici de programare		
	Algoritmi. Analiza algoritmilor: - necesarul de memorie; - complexitatea temporală a algoritmului; - clasificarea algoritmilor. Abordări recursive și iterative.	A35. Utilizarea corectă terminologiei aferente complexității algoritmilor. A36. Estimarea complexității algoritmilor utilizați în procesul de studiu. A37. Estimarea necesarului de memorie necesar pentru execuția unui program. A38. Descrierea structurii unui subprogram recursiv/iterativ. A39. Argumentarea necesității utilizării unui algoritm recursiv/iterativ. A40. Elaborarea algoritmilor recursivi/iterativi pentru problemele din activitatea profesională. A41. Translarea algoritmilor recursivi/iterativi în limbajul de programare. A42. Implementarea algoritmilor recursivi/iterativi în limbajul de programare.
	Metoda trierii. Aranjamente și combinări: - tehnica de triere; - modelul matematic de aranjare a elementelor unei mulțimi; - modelul matematic de combinare a elementelor a două mulțimi; - domeniile de aplicare a metodelor de combinare și aranjare.	A43. Descrierea algoritmului bazat pe metoda trierii. A44. Descrierea modelului matematic pentru aranjamente și combinări. A45. Prezentarea situațiilor de aplicare a metodei trierii. A46. Elaborarea algoritmilor bazați pe metoda trierii conform specificațiilor propuse. A47. Translarea algoritmilor bazat pe metoda trierii în limbajul de programare. A48. Implementarea algoritmilor bazați pe metoda trierii în limbajul de programare.

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități
	Metoda Greedy: - tehnica Greedy; - domeniile de aplicare a metodei Greedy.	A49. Descrierea algoritmului bazat pe metoda Greedy. A50. Prezentarea situațiilor de aplicare a metodei Greedy. A51. Elaborarea algoritmilor bazați pe metoda Greedy conform specificațiilor propuse. A52. Translarea algoritmilor bazați pe metoda Greedy în limbajul de programare. A53. Implementarea algoritmilor bazați pe metoda Greedy în limbajul de programare.
	Metoda reluării: - tehnica de reluire; - domeniile de aplicare a metodei reluării.	A54. Descrierea algoritmului bazat pe metoda reluării. A55. Prezentarea situațiilor de aplicare a metodei reluării. A56. Elaborarea algoritmilor bazați pe metoda reluării conform specificațiilor propuse. A57. Translarea algoritmilor bazați pe metoda reluării în limbajul de programare. A58. Implementarea algoritmilor bazați pe metoda reluării în limbajul de programare.
	Metoda desparte și stăpânește: - tehnica desparte și stăpânește; - domeniile de aplicare a metodei desparte și stăpânește.	A59. Descrierea algoritmului bazat pe metoda desparte și stăpânește. A60. Prezentarea situațiilor de aplicare a metodei desparte și stăpânește. A61. Elaborarea algoritmilor bazați pe metoda desparte și stăpânește conform specificațiilor propuse.

Unități de competență	Unități de conținut	Abilități
		A62. Translarea algoritmilor bazați pe metoda desparte și stăpânește în limbajul de programare. A63. Implementarea algoritmilor bazați pe metoda desparte și stăpânește în limbajul de programare.
	Metoda programării dinamice: - tehnica programării dinamice; - domeniile de aplicare a metodei programării dinamice.	A64. Descrierea algoritmului bazat pe metoda programării dinamice. A65. Prezentarea situațiilor de aplicare a metodei programării dinamice. A66. Utilizarea algoritmilor bazați pe metoda programării dinamice conform specificațiilor propuse. A67. Translarea algoritmilor bazați pe metoda programării dinamice în limbajul de programare.
3. Elemente de teoria grafurilor		
	Grafuri. Terminologie și proprietăți: - graf orientat și neorientat; - adiacență și incidență, grad; - lanț, lanț elementar, drum, drum elementar, ciclu, ciclu elementar, circuit, circuit elementar; - subgraf, graf parțial; - conexitate, tare conexitate, arbore, arbore parțial. Reprezentarea grafurilor în memoria calculatorului: - matrice de adiacență; - matrice de incidență; - liste de adiacență; - lista muchiilor/arcilor.	A68. Utilizarea terminologiei specifice a teoriei grafurilor. A69. Prezentarea problemelor din viața reală, care pot fi soluționate cu ajutorul teoriei grafurilor. A70. Reprezentarea unui graf în memoria calculatorului prin matricea de incidență, matricea de adiacență, liste. A71. Descrierea algoritmilor de introducere, extragere și transformare a diferitelor forme de reprezentare internă a grafurilor A72. Elaborarea algoritmilor bazați pe proprietățile grafurilor. A73. Translarea algoritmilor de introducere, extragere și transformare a diferitelor forme de reprezentare internă a grafurilor în limbajul de programare. A74. Implementarea algoritmilor de introducere, extragere și transformare a diferitelor forme de reprezentare internă a grafurilor în limbajul de programare.

Unități de competență	Unități de conținut	Abilități
	Algoritmi de determinare a drumului minim: - algoritmul Dijkstra; - algoritmul Roy-Floyd; - algoritmul Belman-Ford. Extragerea arborelui minimal într-un graf: - algoritmul Prim; - algoritmul Kruskal.	A75. Prezentarea situațiilor de aplicare drumului minim. A76. Aplicarea algoritmilor de determinare a drumului minim. A77. Prezentarea situațiilor de aplicare a arborelui minimal. A78. Aplicarea algoritmilor de extragere a arborelui minimal. A79. Translarea algoritmilor de determinare a drumului minim în limbajul de programare. A80. Translarea algoritmilor de extragere a arborelui minim în limbajul de programare. A81. Implementarea algoritmilor de determinare a drumului minim în limbajul de programare.

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

Unități de competență	Unități de conținut	Abilități

VII. Studiu individual ghidat de profesor

Materii pentru studiul individual	Produse de elaborat	Modalități de evaluare	Număr de ore
RÎ1. Implementează structuri dinamice de date în aplicații de consolă, gestionând eficient memoria și resursele, conform specificațiilor tehnice.			
Structuri dinamice de date	Set de aplicații de consolă cu utilizarea structurilor dinamice de date	Prezentarea portofoliului. Demonstrarea pe calculator.	12
RÎ4. Dezvoltă algoritmi eficienți pentru reprezentarea și procesarea grafurilor în aplicații de consolă, minimizând consumul de resurse prin metode adecvate de stocare și prelucrare a datelor.			
Tehnici de programare Elemente de teoria grafurilor	Set de aplicații de consolă cu utilizarea tehnicilor de programare și/sau teoriei grafurilor	Prezentarea portofoliului. Demonstrarea pe calculator.	18

VIII. Lucrările practice recomandate

Lucrările practice vor fi efectuate în formă de lucrări de laborator. Tematica lucrărilor recomandate:

1. Prelucrarea tipurilor de date listă.
2. Prelucrarea tipurilor de date stivă coadă.
3. Prelucrarea tipurilor de date arbori binari.
4. Analiza complexității și necesarului de memorie a algoritmilor.
5. Implementarea algoritmilor bazați pe metoda trierii.
6. Implementarea algoritmilor bazați pe metoda Greedy.
7. Implementarea algoritmilor bazați pe metoda reluării.
8. Implementarea algoritmilor bazați pe metoda desparte și stăpânește.
9. Implementarea algoritmilor bazați pe metoda programării dinamice.
10. Introducerea și extragerea grafurilor, transformarea a formelor de reprezentare internă a acestora.
11. Implementarea algoritmilor de determinare a drumurilor minime în grafuri.
12. Implementarea algoritmilor de extragere a arborilor minimali din grafuri.

IX. Sugestii metodologice

Elementele de bază ale Curriculumului sunt competențele ce trebuie formate și dezvoltate în procesul de instruire profesională. Acest scop va fi atins prin organizarea eficientă a procesului de instruire. Pentru aceasta sunt necesare două condiții:

1. Organizarea activităților. Pentru buna organizare a procesului didactic ambii participanți necesită de a-și organiza activitățile. De modul cum sunt organizate acestea depinde în mare măsură nivelul de formare a competențelor. În această ordine de idei, în procesul de organizare a activităților se vor asigura:

- condiții optime pentru buna colaborare dintre elev și profesor;
- un set de procese care duc la îmbunătățirea relațiilor dintre părți;
- un nivel de implicare a părților acționând în baza unor reguli și acțiuni prestabilite.

Activitățile pot fi organizate cât cu prezența fizică, atât și online. Pentru a crea un mediu de învățare cât mai aproape de o clasă reală, este suficient de utilizat trei tipuri de resurse:

O platformă de interacțiune în timp real, cu video și text, cu elevii printre care sunt: Google Meet, Jitsi Meet, Zoom, Discord, Cisco Webex, Microsoft Teams etc.

Aplicații sau platforme de colaborare online, care facilitează schimbul de documente, teste sau teme pentru acasă, între profesori și elevi și înregistrează o evidență a acestora, care permite și feedback din partea profesorului. Aici recomandăm Moodle, Google Classroom și alternative folosite de profesori, precum Edmodo, EasyClass și ClassDojo.

Resurse și aplicații de învățare pe care le poate crea profesorul sau resurse deja existente sub formă de prezentări, lecții, fișe, imagini și clipuri care pot fi folosite atât în timpul lecțiilor live, cât și ca teme de lucru pentru acasă. Aici lista e mai lungă și include aplicații precum TestMoz, LearningApps, Mentimeter, Linolt, ASQ, Kahoot, Quizziz, Wordwall, Padlet, Twinkl sau Digitaliada, precum și surse de inspirație pentru filme, teme și studiu individual.

2. Selectarea adecvată a metodelor de instruire. Se recomandă utilizarea metodelor de instruire precum:

Simularea și modelarea. Simularea este utilizată pentru prezentarea la faza inițială a unor concepte, oferind posibilitatea de ghidare a activității elevului în bază de situații practice. Prin intermediul acestei metode se pot reda, prin analogie, diverse situații, raționamente, care pot să reprezinte relații dintre obiecte, fenomene, procese etc. Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unități de conținut:

- Tipuri dinamice de date;
- Operații specifice tipurilor dinamice de date;
- Arbori binari de căutare.

Problematicizarea mai poate fi denumită și predare prin rezolvare de probleme sau predare productivă de probleme. Conform acestei metode instruitului este pus în fața unor dificultăți create în mod deliberat, și prin depășirea lor învață ceva nou. „Punctul forte” al metodei îl constituie situația-problemă. Din această cauză este necesar de a formula corect situația. La crearea situație de tip problemă se va ține cont de următoarele caracteristici:

- 1) Situația trebuie să prezinte o dificultate pentru instruit, iar pentru a găsi soluția, acesta se va confrunta cu efort de gândire;
- 2) Situația trebuie să prezinte interes, astfel încât acesta să acționeze spre a rezolva problema;
- 3) Situația trebuie să orienteze activitatea instruitului spre a rezolva problema și de al cointeresa pe acesta de a dobândi noi cunoștințe;
- 4) Rezolvarea situației nu va fi posibilă fără a apela la resurselor recent dobândite.

Prin intermediul situației create, instruitul este cointerestat de a studia, analiza și a participa la rezolvarea problemei. Aplicarea acestei metode presupune parcurgerea a patru etape:

- 1) Formularea problemei – este descrisă situația problemă, explicarea, după necesitate a diferitor puncte cheie, care ar permite instruitului să perceapă problema;
- 2) Studierea problemei – se lucrează în mod independent, sunt reactualizate anumite resurse;
- 3) Determinarea soluției – în cadrul acestei etape sunt pregătite resursele necesare, se descoperă mijloacele care duc la rezolvarea problemei și este analizat modul de aplicare a acestora în determinarea soluției;
- 4) Obținerea rezultatului final – se analizează rezultatul obținut și formate anumite concluzii.

Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unități de conținut:

- Analiza algoritmilor;
- Abordări recursive și iterative;
- Metode de programare.

Algoritmizarea reprezintă o metodă de predare-învățare bazată pe utilizarea și valorificarea algoritmilor în procesul de instruire. Algoritmul de instruire se reprezintă sub forma unui grup de scheme, unui set de operații, iar prin parcurgerea lor într-o ordine bine stabilită duce la rezolvarea unui set de probleme caracteristice unei familii de situații. În rezultatul aplicării acestei metode se va oferi posibilitatea elevului de a elabora treptat propriile scheme, aplicabile în diferite circumstanțe didactice. Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unități de conținut:

- Metoda trierii;
- Metoda Greedy;
- Metoda reluării;
- Metoda desparte și stăpânește.

Instruirea asistată de calculator este o metodă didactică care valorifică principiile de modelare și analiză cibernetică. Prin intermediul calculatorului se pune la dispoziția elevului un set de probleme, care necesită a fi analizate, completate sau elaborate. Utilizarea metodei va oferi posibilitatea de organizarea informației conform cerințelor programei adaptabile la capacitățile fiecărui elev; stimularea cognitivă a elevului prin secvențe didactice și întrebări ce vizează depistarea unor lacune, probleme, situații-problemă; rezolvarea sarcinilor didactice prezentate anterior prin reactivarea sau obținerea informațiilor necesare de la resursele informatice apelate prin intermediul calculatorului; realizarea unor sinteze recapitulative după parcurgerea unor teme, module de studiu, lecții; asigurarea unor exerciții suplimentare de stimulare a creativității elevului. Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unități de conținut:

- Metoda programării dinamice;
- Algoritmi de determinare a drumului minim;
- Extragerea arborelui minimal într-un graf.

Metoda studiul de caz valorifică o situație reală care se analizează și se rezolvă. Așa cum problemele rezolvate în stilul orientat pe obiecte au un grad sporit de dificultate, sunt cazuri când este necesar de a prezenta elevului probleme deja rezolvate. Avantajul metodei, constă în faptul că fiecare dintre elev își va aduce aportul la analiza și rezolvarea problemei. În utilizarea acestei metode se conturează câteva etape:

- 1) Selectarea și prezentarea cazului;
- 2) Organizarea echipelor de lucru;

- 3) Prelucrarea și conceptualizarea;
- 4) Structurarea finală a studiului. Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unități de conținut:
 - Structuri dinamice de date prestabilite;
 - Grafuri. Terminologie și proprietăți;
 - Reprezentarea grafurilor în memoria calculatorului.

X. Sugestii de evaluare

Evaluarea competențelor profesionale este procesul prin care sunt colectate și analizate dovezile necesare pentru judecarea competenței în raport cu cerințele calificării profesionale. Calificarea profesională este documentul în care se descriu rezultatele învățării în concordanță cu cerințele pieței muncii, specificate în standardul ocupațional/ profilul ocupațional. Evaluarea competențelor profesionale este un proces complet diferit de sistemul tradițional de evaluare a cunoștințelor. Evaluarea competențelor profesionale este un proces care presupune consultarea și colaborarea dintre elev și profesor. Evaluarea competențelor are loc prin furnizarea de către elev a dovezilor de competență care sunt interpretate de către profesor. Dovezile de competență acumulate sunt rezultate considerate parțiale și atât elevul cât și profesorul pot solicita clarificări suplimentare.

Procedura de evaluare a competențelor profesionale pentru modulul *Programarea calculatorului*, va oferi elevilor posibilitatea de a-și demonstra atât cunoștințele teoretice și practice. Metodele folosite în procesul de evaluare vor evidenția cunoștințele și deprinderile necesare pentru efectuarea activităților de muncă și, mai ales, capacitatea elevului de a obține rezultatele practice așteptate.

Activitățile de evaluare vor fi orientate spre motivarea elevilor și obținerea unui feedback continuu, fapt ce va permite corectarea operativă a procesului de învățare, stimularea autoevaluării și a evaluării reciproce, evidențierea succeselor, implementarea evaluării selective sau individuale. Pentru a eficientiza procesele de evaluare, înainte de a demara evaluările, cadrul didactic va aduce la cunoștință elevilor tematica lucrărilor, modul de evaluare (bareme/grile/criterii de notare) și condițiile de realizare a fiecărei evaluări.

Evaluarea curentă/formativă se va realiza prin diverse modalități: observarea comportamentului elevului, analiza rezultatelor activității elevului, discuția/conversația, prezentarea proiectelor individuale de activitate. Prin evaluarea curentă/formativă, cadrele didactice informează elevul despre nivelul de performanță; îl motivează să se implice în dobândirea competențelor profesionale.

Evaluarea sumativă se realizează la finele modulului în baza simulării în atelier a unei situații de problemă din contexte profesionale variate, care solicită elevului demonstrarea competenței profesionale. Cadrele didactice vor elabora sarcini prin care vor orienta comportamentul profesional al elevului spre demonstrarea sistemului de cunoștințe și abilități. În acest scop, vor fi clar stabiliți indicatorii și descriptorii de performanță ai procesului și produsului realizat de către elev.

Portofoliul reprezintă o metodă complexă de evaluare în care un rezultat al evaluării este elaborat pe baza aplicării unui ansamblu variat de probe și instrumente de evaluare. Portofoliul, de regulă este realizat pe o perioadă mai îndelungată (în decursul mai multor ore). Conținutul unui portofoliu este reprezentat de rezultatele la: lucrări practice, studiul individual, investigații, referate și proiecte, observarea sistematică la clasă, autoevaluarea elevului, chestionare de atitudini etc. Alegerea elementelor ce formează portofoliul este realizată de către profesor (astfel încât acestea să ofere informații concludente

privind pregătirea, evoluția, atitudinea elevului) sau chiar de către elev (pe considerente de performanță, preferințe etc.). Structurarea evaluării sub forma de portofoliu se dovedește deosebit de utilă, atât pentru profesor, cât și pentru elev sau părinții acestuia. Pentru a realiza o evaluare pe bază de portofoliu, profesorul:

- va comunica elevilor intenția de a realiza un portofoliu, adaptând instrumentele de evaluare ce constituie “centrul de greutate” ale portofoliului la specificul unității de învățare;
- va alege componentele ce formează portofoliul, dând și elevului posibilitatea de a adăuga piese pe care le consideră relevante pentru activitatea sa;
- va evalua separat fiecare piesă a portofoliului în momentul realizării ei, dar va asigura și un sistem de criterii pe baza cărora să realizeze evaluarea globală și finală a portofoliului;
- va pune în evidență evoluția elevului, particularitățile de exprimare și de raportare a acestuia la aria vizată;
- va integra rezultatul evaluării portofoliului în sistemul general de notare.

Competențele elevului se manifestă prin produse concrete, care sunt analizate de către profesor în raport cu aspectele critice stabilite pentru unitate/unitățile de competență pentru care este evaluat. Dovezile de competență sunt informațiile produse de un elev din care rezultă că îndeplinește toate aspectele descrise de unitatea/unitățile de competență pentru care este evaluat, respectiv are cunoștințele și deprinderile necesare.

Evaluarea nivelului de dezvoltare a competențelor în cadrul orelor:

- **teoretice** se va realiza prin teste, exemple de aplicare a cunoștințelor teoretice în practică, machete etc.;
- **de laborator** se va realiza prin elaborarea de către elev, în termeni concreți, a aplicațiilor web având la bază unitățile de conținut studiate în cadrul orelor teoretice precum și abilitățile anterior dezvoltate;
- **de studiu individual** se va realiza prin studierea de către elev a materialelor suplimentare decât cele oferite în cadrul orelor de tip contact direct și prezentarea de portofolii pentru anumite unități de conținut și aplicații web complexe prin care elevul își va demonstra abilitățile formate.

Probe de evaluare a competențelor, în baza situațiilor de problemă de la viitoarele locuri de muncă:

- elaborarea aplicațiilor de consolă conform specificațiilor propuse;
- evaluarea aplicațiilor de consolă elaborate;
- selectarea structurii dinamice de date conform specificațiilor tehnice;
- elaborarea algoritmilor conform metodei;
- modificarea aplicațiilor de consolă conform specificațiilor propuse;
- testarea aplicațiilor de consolă elaborate.

În calitate de **produse pentru măsurarea competențelor** se vor folosi:

- aplicații de consolă elaborate conform specificațiilor propuse;

- algoritmi elaborați conform specificațiilor propuse;
- structuri dinamice de date gestionate conform specificațiilor propuse.

Criteriile de evaluare a produselor pentru măsurarea competenței vor include:

- Utilizarea corectă a instrucțiunilor limbajului de programare.
- Corectitudinea algoritmilor elaborați.
- Fundamentarea deciziilor.
- Ținuta lingvistică.
- Respectarea termenilor de elaborare.

Forma finală de evaluare a unității de curs este examenul. Pentru a fi admis la examen elevul trebuie să susțină evaluările curente, să însușească materialul teoretic și să îndeplinească sarcinile propuse pentru studiul individual. Nota finală va fi formată: 60% - media notelor curente și 40% - nota de la examen.

XI. Resurse necesare pentru atingerea rezultatelor învățării

Pentru realizarea competențelor specifice cursului de „Programarea calculatorului”, laboratorul trebuie să fie dotat cu echipamente tehnologice moderne, mobilier ergonomic. Echipamentele de bază includ calculatoare performante, dotate cu software de dezvoltare actualizat, precum și dispozitive periferice esențiale (monitoare, tastaturi, mouse-uri). Spațiul trebuie să fie bine iluminat, ventilat și să respecte normele de ergonomie, pentru a asigura confortul elevilor pe durata lucrărilor practice. Întreținerea echipamentelor implică actualizarea periodică a software-ului, verificarea funcționării hardware-ului și curățarea regulată a componentelor. De asemenea, este necesară gestionarea corectă a cablurilor și a conexiunilor pentru a preveni accidentele. Utilizarea resurselor trebuie să respecte normele de siguranță electrică și să includă măsuri de protecție împotriva descărcărilor electrostatice.

Standardul de dotare a laboratoarelor

Suprafața totală a laboratorului – min. 30 m²

Suprafața pentru un elev – min. 2 m²

Numărul locurilor de lucru – min. 15

Nr. crt.	Denumirea	Cantitatea per elev	Cantitatea per atelier
a) Echipamente			
1.	Calculator cu specificații minime: - Procesor Minim 4 nuclee, frecvență de 2,5 - 3,0 GHz - Memorie operativă (RAM) Minim 16 GB - Unitate de stocare SSD cu capacitatea minimă de 500 GB (recomandat 1 TB pentru proiecte mari) - Ecran Diagonală minimă de 24", rezoluție Full HD (1920x1080)	1	15

Nr. crt.	Denumirea	Cantitatea per elev	Cantitatea per atelier
	- Placă de rețea Ethernet Gigabit (1 Gb), Wi-Fi standard AC sau AX Tastatură și mouse Ergonomice, cu funcționalitate completă		
2.	Videoproiector cu ecran de proiecție	-	1
3.	Tabla albă / tabla interactivă	-	1
b) Mobilier și tehnică sanitară			
4.	Masa de lucru / Birou pentru elev	1	15
5.	Scaun ergonomic ajustabil	1	15
6.	Sistem de ventilație și aer condiționat	-	1
c) Utilaj tehnologic			
7.	Router Wi-Fi de mare viteză	-	1
d) Instrumente și dispozitive			
8.	Softuri licențiate pentru dezvoltare (Visual Studio, Visual Studio Code, CodeBlocks, Dev C++ etc.)	-	15
9.	Platforme educaționale	-	1
e) Inventar și ustensile			
10.	Manuale și ghiduri de programare	1	15
11.	Markere pentru tablă albă	-	4
12.	Burete pentru tablă albă	-	1

XII. Resursele didactice recomandate elevilor

Nr. crt.	Denumirea resursei	Locul în care poate fi consultată/ accesată resursa
1.	Emanuela Cerchez, Marinel-Paul Șerban "Programarea în limbajul C/C++ pentru liceu." Volumul al II-lea: Metode și tehnici de programare POLIROM 2005	Biblioteca
2.	Emanuela Cerchez, Marinel-Paul Șerban "Programarea în limbajul C/C++ pentru liceu". Teoria grafurilor Structuri de date dinamice (liniare și arborescente) Pattern-matching Hashing Geometrie computațională" Volumul III POLIROM	Biblioteca
3.	A. Ruceanu, Proiectarea algoritmilor.	http://www.ruceanu.ro/adrian/wp-content/cursuri/pa2014.php
4.	G. Vasilache, S. Gîncu Culegere de probleme la informatica, Chișinău, 2012.	http://en.calameo.com/read/002801569a611d413be1c

5.	Algoritmi și structuri de date. Note de Curs.	https://ru.scribd.com/document/103258372/Algoritmi-Si-Structuri-de-Date
6.	S. Cataranciuc, TEORIA GRAFURILOR IN PROBLEME SI APLICATII, Chișinău, 2004.	http://www.math.md/studlib/matematika/teoria_graf.html
7.	Laborator 8: Drumuri minime.	http://elf.cs.pub.ro/pa/wiki/laboratoare/laborator-08
8.	Curs online: C++ Essentials 2	https://www.netacad.com/courses/c-plus-plus-essentials-2?courseLang=en-US