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

"Aprob"
Directorul Centrului de Excelență în Informatică şi Tehnologii Informaționale
_____ Vitalie Zavadschi

Curriculumul modular
F.03.O.013 Programarea calculatorului

Specialitatea**: 61210 Administrarea aplicațiilor web**
Calificarea: **Tehnician de site-uri web**

Chișinău 2022

Aprobat de:

Consiliul metodico-ştiinţific al Centrului de Excelenţă în Informatică şi Tehnologii Informaţionale, proces verbal Nr.3 din 31.10.2022

Director adjunct _____
Obadă Liuba

Autori:

Gîncu Silviu, doctor în pedagogie, grad didactic superior.
Șarapanovscaia Irina, grad didactic doi, Centrul de Excelență în Informatică și Tehnologii Informaționale

Actualizat în baza Planului de învățământ nr. SC-15/20, aprobat prin OMEC Nr.1310, din 23.11.2020.

Recenzenți:

1. *Nastasenco Veaceslav*, director ”Allied Testing – Moldova” SRL, adresa: str. Ismail 33/1, mun.Chișinău.
2. *Țugui Mihai*, director ”EBS Integrator” SRL, adresa: str. Ion Inculeț 33, mun. Chișinău.

Adresa Curriculumului în Internet:

https://mecc.gov.md/ro/content/curriculum-invatamintul-profesional-tehnic-postsecundar

Cuprins

I. Preliminarii.....4

II. Motivaţia, utilitatea modulului pentru dezvoltarea profesională.....4

III. Competenţele profesionale specifice modulului.....5

IV. Administrarea modulului.....5

V. Unităţile de învățare.....6

VI. Repartizarea orientativă a orelor pe unități de învățare.....11

VII. Studiu individual ghidat de profesor.....11

VIII. Lucrările practice recomandate.....11

IX. Sugestii metodologice.....12

X. Sugestii de evaluare a competențelor profesionale.....14

XI. Resursele necesare pentru desfășurarea procesului de studii.....16

XII. Resursele didactice recomandate elevilor.....17

I. Preliminarii

Cursul Programarea calculatorului conține noțiunile fundamentale privind structurile dinamice de date, algoritmi, tehnici de programare necesare pentru elaborarea unui software performant şi portabil.

Statutul Curriculumului. Curriculumul modular "Programarea calculatorului" este un document normativ şi obligatoriu pentru realizarea procesului de pregătire a tehnicienilor în învățământul profesional tehnic post secundar, care vor elabora diverse aplicații în conformitate cu sarcinile de lucru.

Funcțiile Curriculumului. Funcțiile de bază ale Curriculumului sunt:

- act normativ al procesului de predare, învățare, evaluare şi certificare în contextul unei pedagogii axate pe competențe;
- reper pentru proiectarea didactică şi desfășurarea procesului educațional din perspectiva unei pedagogii axate pe competențe;
- componentă de bază pentru elaborarea strategiei de evaluare şi 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 instituţiile de învăţământ profesional tehnic post secundar;
- autorilor de manuale şi ghiduri metodologice;
- elevilor care îşi fac studiile la specialitatea în cauză;
- membrilor comisiilor pentru examenele de calificare;
- membrilor comisiilor de identificare, evaluare şi recunoaştere a rezultatelor învăţării, dobândite în contexte non-formale şi informale.

Scopul studierii acestui modul constă în formarea şi dezvoltarea competenţei profesionale specifice de utilizarea a tipurilor dinamice de date și a tehnicilor de programare în mentenanţa şi actualizarea produselor-program (softurilor) de sistem precum şi de aplicaţii. De asemenea, modulul contribuie la dezvoltarea competenţei profesionale generale de respectare şi de promovare a normelor de drept informatic.

Modulul în cauză poate fi studiat după însuşirea în mod obligatoriu a următoarelor unităţi de curs:

F.01.O.010 Programarea structurată.

F.02.O.012 Programarea procedurală.

II. Motivaţia, utilitatea modulului pentru dezvoltarea profesională

Studierea acestui modul va contribui la formarea şi dezvoltarea de competente profesionale ce corespund nivelului patru de calificare:

- cunoștințe faptice, principii, procese și concepte generale din domeniul elaborării produselor program;
- abilități cognitive și practice necesare pentru elaborarea aplicațiilor de consolă conform tematicilor incluse;
- asumarea responsabilității pentru mentenanţa de aplicaţii.

Competențele formate şi dezvoltate în cadrul acestui modul vor fi necesare pentru studierea următoarelor module:

F.04.O.015 Asistență pentru programarea orientată pe obiecte

S.04.A.032 Asistență pentru programarea vizuală

S.05.O.019 Asistență pentru crearea site-urilor WEB

S.06.O.020 Asistență pentru programare client-side a site-urilor Web

S.07.O.021 Asistență pentru programare server-side a site-urilor Web

S.08.O.026 Programarea aplicațiilor pentru dispozitive mobile

S.08.A.038 Dezvoltarea aplicațiilor desktop

De asemenea, ele vor fi de un real folos în activitatea profesională a tehnicianului, în special, în ocupațiile legate de gestiunea produselor-program utilizate în companii..

III. Competenţele profesionale specifice modulului

În cadrul modulului vor fi formate şi dezvoltate următoarele competențe profesionale specifice:

CS1. Prelucrarea tipurilor dinamice de date în cadrul aplicațiilor de consolă.

CS2. Utilizarea structurilor dinamice de date pentru problemele întâlnite în activitatea profesională.

CS3. Gestionarea eficientă a memoriei interne a calculatorului.

CS4. Utilizarea tehnicilor de programare pentru problemele întâlnite în activitatea profesională.

CS5. Utilizarea grafurilor pentru problemele întâlnite în activitatea profesională.

CS6. Alegerea tehnicii de programare adecvate problemei.

IV. Administrarea modulului

Semestrul	Numărul de ore				Forma de evaluare	Numărul de credite
	Total	Contact direct		Ore de studiu individual		
		Teorie	Laborator			
III	90	30	30	30	Examen	3

V. Unităţile de învăţare

Unităţi de competenţă	Unităţi de conţinut	Abilităţi
	1. **Structuri dinamice de date**	
UC1. Prelucrarea tipurilor dinamice de date de tip listă în cadrul aplicațiilor de consolă	1. Structuri dinamice de date. Tipuri dinamice de date: - listă simplu înlănțuite; - liste dublu înlănțuite; - liste circulare. 2. 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	Unităţi de conținut	Abilități
UC2. Prelucrarea tipurilor dinamice de date de tip stivă/coadă în cadrul aplicațiilor de consolă	3. Structuri dinamice de date prestabilite: - stivă; - coadă. 4. Operații specifice tipurilor dinamice de date: - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor.	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. 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.
UC3. Prelucrarea tipurilor dinamice de date de tip arbore binar în cadrul aplicațiilor de consolă	5. Arbori binari. Operații specifice tipului de date arbori binari: - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor. 6. Arbori binari de căutare. Operații specifice: - creare; - adăugare; - excludere;	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.

Unități de competență	Unităţi de conținut	Abilități
	- parcurgere; - căutare; - distrugere.	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.
2. **Tehnici de programare**		
UC4. Estimarea complexității algoritmilor	7. Algoritmi. Analiza algoritmilor: - necesarul de memorie; - complexitatea temporală a algoritmului; - clasificarea algoritmilor. 8. 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.
UC5. Utilizarea algoritmilor bazați pe metoda trierii în cadrul aplicațiilor de consolă	9. Metoda trierii. Aranjamente şi 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 şi aranjare.	A43. Descrierea algoritmului bazat pe metoda trierii. A44. Descrierea modelului matematic pentru aranjamente şi 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	Unităţi de conținut	Abilități
UC6. Utilizarea algoritmilor bazați pe metoda Greedy în cadrul aplicațiilor de consolă	10. 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.
UC7. Utilizarea algoritmilor bazați pe metoda reluării în cadrul aplicațiilor de consolă	11. Metoda reluării: - tehnica de reluare; - 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.
UC8. Utilizarea algoritmilor bazați pe metoda desparte şi stăpânește în cadrul aplicațiilor de consolă	12. Metoda desparte şi stăpânește: - tehnica desparte şi stăpânește; - domeniile de aplicare a metodei desparte şi stăpânește.	A59. Descrierea algoritmului bazat pe metoda desparte şi stăpânește. A60. Prezentarea situațiilor de aplicare a metodei desparte şi stăpânește. A61. Elaborarea algoritmilor bazați pe metoda desparte şi stăpânește conform specificațiilor propuse. A62. Translarea algoritmilor bazați pe metoda desparte şi stăpânește în limbajul de programare. A63. Implementarea algoritmilor bazați pe metoda desparte şi stăpânește în limbajul de programare.
UC9. Utilizarea algoritmilor bazați pe metoda programării dinamice în cadrul aplicațiilor de consolă	13. 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.

Unități de competență	Unităţi de conținut	Abilități
		A67. Translarea algoritmilor bazați pe metoda programării dinamice în limbajul de programare.
3. **Grafuri**		
UC10. Utilizarea algoritmilor bazați pe metode de reprezentare a grafurilor în cadrul aplicațiilor de consolă	14. Grafuri. Terminologie şi proprietăți. - graf orientat şi 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. 15. Reprezentarea grafurilor în memoria calculatorului. - matrice de adiacență, - matrice de incidență - liste de adiacență, - lista muchiilor/arcelor.	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, listei. 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.
UC11. Utilizarea algoritmilor de calcul a drumului minim într-un graf	16. Algoritmi de determinare a drumului minim - algoritmul Dijkstra; - algoritmul Roy-Floyd; - algoritmul Belman-Ford. 17. 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.

VI. Repartizarea orientativă a orelor pe unități de învățare

Nr. crt.	Unități de învățare	Numărul de ore			
		Total	Contact direct		Ore de studiu individual
			Teorie	Laborator	
1.	Structuri dinamice de date	36	12	12	12
2.	Tehnici de programare	36	12	12	12
3.	Grafuri	18	6	6	6
	Total	**90**	**30**	**30**	**30**

VII. Studiu individual ghidat de profesor

Materii pentru studiul individual	Produse de elaborat	Modalităţi de evaluare	Termeni de realizare
Structuri dinamice de date			
1. Structuri dinamice de date	Portofoliu: Set de aplicații de consolă cu utilizarea structurilor dinamice de date	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 6
Tehnici de programare			
2. Tehnici de programare	Portofoliu: Set de aplicații de consolă cu utilizarea tehnicilor de programare	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 12
Grafuri			
3. Grafuri	Portofoliu: Set de aplicații de consolă cu utilizarea grafurilor.	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 15

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 şi 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 şi dezvoltate în procesul de instruire profesională. Aceste 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-şi 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 şi 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, Mentimetr, LinoIt, 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 şi 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 unităţi de conținut:

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

Problematizarea mai poate fi denumită şi predare prin rezolvare de probleme sau predare productivă de probleme. Conform acestei metode instruitului este pus în fața unor dificultăți create în mod deliberat, şi 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:

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

Prin intermediul situației create, instruitul este cointeresat de a studia, analiza şi 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 şi este analizat modul de aplicare a acestora în determinarea soluției;
4. Obținerea rezultatului final – se analizează rezultatul obținut şi formate anumite concluzii.

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

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

Algoritmizarea reprezintă o metodă de predare-învățare bazată pe utilizarea şi 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 unităţi de conținut:

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

Instruirea asistată de calculator este o metodă didactică care valorifică principiile de modelare şi 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 şi 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 unităţi 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 şi 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 îşi va aduce aportul la analiza şi rezolvarea problemei. În utilizarea acestei metode se conturează câteva etape: 1) Selectarea şi prezentarea cazului; 2) Organizarea echipelor de lucru; 3) Prelucrarea şi conceptualizarea; 4) Structurarea finală a studiului. Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unităţi de conținut:

- Structuri dinamice de date prestabilite;
- Grafuri. Terminologie şi proprietăți;
- Reprezentarea grafurilor în memoria calculatorului.

X. Sugestii de evaluare a competențelor profesionale

Evaluarea competențelor profesionale este procesul prin care sunt colectate şi analizate dovezile necesare pentru judecarea competenţei în raport cu cerințele calificării profesionale. Calificarea profesională este documentul în care se descriu rezultatele învăţării 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 şi colaborarea dintre elev şi profesor. Evaluarea competențelor are loc prin furnizarea de către elev a dovezilor de competentă care sunt interpretate de către profesor. Dovezile de competență acumulate sunt rezultate considerate parțiale şi 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-şi demonstra atât cunoştinţele teoretice şi practice. Metodele folosite în procesul de evaluare vor evidenția cunoștințele şi deprinderile necesare pentru efectuarea activităților de muncă şi, mai ales, capacitatea elevului de a obține rezultatele practice aşteptate.

Activitățile de evaluare vor fi orientate spre motivarea elevilor şi obținerea unui feedback continuu, fapt ce va permite corectarea operativă a procesului de învățare, stimularea autoevaluării şi 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) şi 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 competenţei profesionale. Cadrele didactice vor elabora sarcini prin care vor orienta comportamentul profesional al elevului spre demonstrarea sistemului de cunoştinţe şi abilități. În acest scop, vor fi clar stabiliți indicatorii şi descriptorii de performanță ai procesului şi 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 şi 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 competenţei 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. Resursele necesare pentru desfășurarea procesului de studii

Cerinţe faţă de sălile de curs	
Pentru orele teoretice	Cabinet de informatică cu 15 calculatoare Proiector
Pentru orele de laborator	Laborator de informatică care asigură fiecărui elev un calculator
Cerinţe pentru lecţii online	
Pentru orele teoretice şi de laborator	Instrumente de comunicare sincronă Platforme de colaborare online Instrumente digitale de evaluare
Cerinţe tehnice	
Parametri tehnici minimi ale calculatorului	**Procesor**: 2 GHz **Memorie operativă**: 4 GB **Unitate de stocare**: 500 GB **Afișaj și grafică**: size: 22'', resolution: 1366x768 **Network**: Ethernet, 100 Mb
Software	Sistem de Operare Microsoft Windows Code::Blocks Dev C/CPP Visual Studio 2019

XII. Resursele didactice recomandate elevilor

Nr. crt.	Denumirea resursei	Locul în care poate fi consultată/ accesată resursa
1.	A. Ruceanu, Proiectarea algoritmilor.	http://www.runceanu.ro/adrian/wp-content/cursuri/pa2014.php
2.	G. Vasilache, S. Gîncu Culegere de probleme la informatica, Chișinău, 2012.	http://en.calameo.com/read/002801569a611d413be1c
3.	Algoritmi și structuri de date. Note de Curs.	https://ru.scribd.com/document/103258372/Algoritmi-Si-Structuri-de-Date
4.	S. Cataranciuc, TEORIA GRAFURILOR IN PROBLEME SI APLICATII, Chișinău, 2004.	http://www.math.md/studlib/matematica/teoria_graf.html
5.	Laborator 8: Drumuri minime.	http://elf.cs.pub.ro/pa/wiki/laboratoare/laborator-08