



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

"Aprob"

Directorul Centrului de Excelență în
Informatică și Tehnologii Informaționale




Vitalie Zavadschi

20 decembrie 2016

Curriculumul modular
F.03.O.014 Programarea calculatorului

Specialitatea: 61230 Rețele de calculatoare
Calificarea: Tehnician pentru rețele de calculatoare

Chișinău 2016

Curriculumul a fost elaborat în cadrul Proiectului *EuropeAid/133700/C/SER/MD/12*
"Asistență tehnică pentru domeniul învățământ și formare profesională
în Republica Moldova",
implementat cu suportul financiar al Uniunii Europene



Autori:

Gîncu Silviu, doctor în pedagogie, grad didactic superior.

Șarapanovscaia Irina, grad didactic doi, Centrul de excelență în informatică și tehnologii informaționale.

Aprobat de:

Consiliul metodic-științific al Centrului de Excelență în Informatică și Tehnologii Informaționale.



Director

Vitalie Zavadschi

20 decembrie 2016

Recenzenți:

1. Asociația Națională a Companiilor din Domeniul TIC/ATIC, adresa: str.Maria Cibotari 28, mun.Chișinău, director executiv Chirița Ana.
2. „EBS Integrator” SRL, adresa: str.Ion Inculeț 33, mun.Chișinău, director Aremesu Vitalie.

Adresa Curriculumului în Internet:

Portalul național al învățământului profesional tehnic

<http://www.ipt.md/ro/produse-educationale>.

Cuprins

I. Preliminarii	4
II. Motivația, utilitatea modulului pentru dezvoltarea profesională.....	5
III. Competențele profesionale specifice modulului	5
IV. Administrarea modulului	5
V. Unitățile de învățare	6
VI. Repartizarea orientativă a orelor pe unități de învățare	13
VII. Studiu individual ghidat de profesor	13
VIII. Lucrările practice recomandate	14
IX. Sugestii metodologice	15
X. Sugestii de evaluare a competențelor profesionale	17
XI. Resursele necesare pentru desfășurarea procesului de studii	19
XII. Resursele didactice recomandate elevilor	20

I. Preliminarii

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

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

Funcțiile Curriculumului. Funcțiile de bază ale Curriculumul 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 postsecundar;
- 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 studierii acestui modul constă în formarea și dezvoltarea competenței profesionale specifice de utilizarea a tipurilor dinamice de date și a tehnicilor de programare în mentenanța și actualizarea produselor-program (softurilor) de sistem precum și de aplicații. De asemenea, modulul contribuie la dezvoltarea competenței profesionale generale de respectare și de promovare a normelor de drept informatic.

Modulul în cauză poate fi studiat după însușirea în mod obligatoriu a următoarelor unități de curs:

- Programarea structurată.
- Programarea procedurală.

II. Motivația, utilitatea modulului pentru dezvoltarea profesională

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

- cunoștințe factice, 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 mentenanța de aplicații.

Competențele formate și dezvoltate în cadrul acestui modul vor fi necesare pentru studierea unităților de curs orientate spre elaborarea/dezvoltarea produselor program. 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. Competențele profesionale specifice modulului

În cadrul modulului vor fi formate și 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				Modalitatea de evaluare	Numărul de credite
	Total	Contact direct		Lucrul individual		
		Prelegeri	Practică/ Seminar			
III	90	30	30	30	examen	3

V. Unitățile de învățare

Unități de competență	Unități de conținut	Abilități
1. Structuri dinamice de date		
<p>UC1. Prelucrarea tipurilor dinamice de date de tip listă în cadrul aplicațiilor de consolă</p>	<p>1. Structuri dinamice de date. Tipuri dinamice de date:</p> <ul style="list-style-type: none"> - listă simplu înlănțuite; - liste dublu înlănțuite; - liste circulare. <p>2. Operații specifice tipurilor dinamice de date:</p> <ul style="list-style-type: none"> - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor. 	<p>A1. Declararea unei structuri dinamice de date.</p> <p>A2. Alocarea dinamică a memoriei unei variabile dinamice.</p> <p>A3. Eliberarea memoriei dinamice alocate unei variabile dinamice.</p> <p>A4. Crearea unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p> <p>A5. Afișarea datelor unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p> <p>A6. Căutarea datelor unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p> <p>A7. Permutarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p> <p>A8. Eliminarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p> <p>A9. Adăugarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p> <p>A10. Interclasarea structurilor dinamice de date.</p> <p>A11. Ordonarea datelor în cadrul unei structuri dinamice de date de tip listă, conform specificațiilor propuse.</p>

Unități de competență	Unități de conținut	Abilități
		<p>A12. Distrugerea unei structuri dinamice de date de tip listă.</p> <p>A13. Elaborarea algoritmilor pentru tipuri de date de tip listă.</p> <p>A14. Translarea algoritmilor pentru tipuri de date de tip listă în limbajul de programare.</p> <p>A15. Implementarea algoritmilor pentru tipuri de date de tip listă în limbajul de programare.</p>
<p>UC2. Prelucrarea tipurilor dinamice de date de tip stivă/coadă în cadrul aplicațiilor de consolă</p>	<p>3. Structuri dinamice de date prestabilite:</p> <ul style="list-style-type: none"> - stivă; - coadă. <p>4. Operații specifice tipurilor dinamice de date:</p> <ul style="list-style-type: none"> - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor. 	<p>A16. Crearea unei structuri dinamice de date de tip stivă, coadă.</p> <p>A17. Afișarea datelor unei structuri dinamice de date de tip stivă, coadă.</p> <p>A18. Căutarea datelor unei structuri dinamice de date de tip stivă, coadă.</p> <p>A19. Eliminarea datelor în cadrul unei structuri dinamice de date de tip stivă, coadă.</p> <p>A20. Adăugarea datelor în cadrul unei structuri dinamice de date de tip stivă, coadă.</p> <p>A21. Distrugerea unei structuri dinamice de date de tip stivă, coadă.</p> <p>A22. Elaborarea algoritmilor pentru tipuri de date de tip stivă, coadă.</p> <p>A23. Translarea algoritmilor pentru tipuri de date de tip stivă, coadă în limbajul de programare.</p> <p>A24. Implementarea algoritmilor pentru tipuri de date de tip stivă, coadă în limbajul de programare.</p>

Unități de competență	Unități de conținut	Abilități
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: <ul style="list-style-type: none"> - creare; - adăugare; - excludere; - parcurgere; - căutare; - distrugere; - extragerea datelor. 6. Arbori binari de căutare. Operații specifice: <ul style="list-style-type: none"> - 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.
2. Tehnici de programare		
UC4. Estimarea complexității algoritmilor	7. Algoritmi. Analiza algoritmilor: <ul style="list-style-type: none"> - necesarul de memorie; - complexitatea temporală a algoritmului; 	A35. Utilizarea corectă terminologiei aferente complexității algoritmilor. A36. Estimarea complexității algoritmilor utilizați în procesul

Unități de competență	Unități de conținut	Abilități
	<ul style="list-style-type: none"> - clasificarea algoritmilor. <p>8. Abordări recursive și iterative</p>	<p>de studiu.</p> <p>A37. Estimarea necesarului de memorie necesar pentru execuția unui program.</p> <p>A38. Descrierea structurii unui subprogram recursiv/iterativ.</p> <p>A39. Argumentarea necesității utilizării unui algoritm recursiv/iterativ.</p> <p>A40. Elaborarea algoritmilor recursivi/iterativi pentru problemele din activitatea profesională.</p> <p>A41. Translarea algoritmilor recursivi/iterativi în limbajul de programare.</p> <p>A42. Implementarea algoritmilor recursivi/iterativi în limbajul de programare.</p>
UC5. Utilizarea algoritmilor bazați pe metoda trierii în cadrul aplicațiilor de consolă	<p>9. Metoda trierii. Aranjamente și combinații:</p> <ul style="list-style-type: none"> - 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. 	<p>A43. Descrierea algoritmului bazat pe metoda trierii.</p> <p>A44. Descrierea modelului matematic pentru aranjamente și combinații.</p> <p>A45. Prezentarea situațiilor de aplicare a metodei trierii.</p> <p>A46. Elaborarea algoritmilor bazați pe metoda trierii conform specificațiilor propuse.</p> <p>A47. Translarea algoritmilor bazat pe metoda trierii în limbajul de programare.</p> <p>A48. Implementarea algoritmilor bazați pe metoda trierii în limbajul de programare.</p>
UC6. Utilizarea algoritmilor bazați pe metoda Greedy în	<p>10. Metoda Greedy:</p> <ul style="list-style-type: none"> - tehnica Greedy; 	<p>A49. Descrierea algoritmului bazat pe metoda Greedy.</p> <p>A50. Prezentarea situațiilor de aplicare a metodei Greedy.</p>

Unități de competență	Unități de conținut	Abilități
<p>cadrul aplicațiilor de consolă</p>	<ul style="list-style-type: none"> - domeniile de aplicare a metodei Greedy. 	<p>A51. Elaborarea algoritmilor bazați pe metoda Greedy conform specificațiilor propuse.</p> <p>A52. Translarea algoritmilor bazați pe metoda Greedy în limbajul de programare.</p> <p>A53. Implementarea algoritmilor bazați pe metoda Greedy în limbajul de programare.</p>
<p>UC7. Utilizarea algoritmilor bazați pe metoda reluării în cadrul aplicațiilor de consolă</p>	<p>11. Metoda reluării:</p> <ul style="list-style-type: none"> - tehnica de reluare; - domeniile de aplicare a metodei reluării. 	<p>A54. Descrierea algoritmului bazat pe metoda reluării.</p> <p>A55. Prezentarea situațiilor de aplicare a metodei reluării.</p> <p>A56. Elaborarea algoritmilor bazați pe metoda reluării conform specificațiilor propuse.</p> <p>A57. Translarea algoritmilor bazați pe metoda reluării în limbajul de programare.</p> <p>A58. Implementarea algoritmilor bazați pe metoda reluării în limbajul de programare.</p>
<p>UC8. Utilizarea algoritmilor bazați pe metoda desparte și stăpânește în cadrul aplicațiilor de consolă</p>	<p>12. Metoda desparte și stăpânește:</p> <ul style="list-style-type: none"> - tehnica desparte și stăpânește; - domeniile de aplicare a metodei desparte și stăpânește. 	<p>A59. Descrierea algoritmului bazat pe metoda desparte și stăpânește.</p> <p>A60. Prezentarea situațiilor de aplicare a metodei desparte și stăpânește.</p> <p>A61. Elaborarea algoritmilor bazați pe metoda desparte și stăpânește conform specificațiilor propuse.</p> <p>A62. Translarea algoritmilor bazați pe metoda desparte și stăpânește în limbajul de programare.</p> <p>A63. Implementarea algoritmilor bazați pe metoda desparte și stăpânește în limbajul de programare.</p>

Unități de competență	Unități de conținut	Abilități
UC9. Utilizarea algoritmilor bazați pe metoda programării dinamice în cadrul aplicațiilor de consolă	13. Metoda programării dinamice: <ul style="list-style-type: none"> - 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. Grafuri		
UC10. Utilizarea algoritmilor bazați pe metode de reprezentare a grafurilor în cadrul aplicațiilor de consolă	14. Grafuri. Terminologie și proprietăți. <ul style="list-style-type: none"> - 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. 15. Reprezentarea grafurilor în memoria calculatorului. <ul style="list-style-type: none"> - 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, 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.

Unități de competență	Unități de conținut	Abilități
UC11. Utilizarea algoritmilor de calcul a drumului minim într-un graf	16. Algoritmi de determinare a drumului minim <ul style="list-style-type: none"> - algoritmul Dijkstra; - algoritmul Roy-Floyd; - algoritmul Belman-Ford. 17. Extragerea arborelui minimal într-un graf: <ul style="list-style-type: none"> - 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 minimal în limbajul de programare. A81. Implementarea algoritmilor de determinare a drumului minim în limbajul de programare. A82. Implementarea algoritmilor de extragere a arborelui minimal în limbajul de programare.

VI. Repartizarea orientativă a orelor pe unități de învățare

Nr. crt.	Unități de învățare	Numărul de ore			
		Total	Contact direct		Lucrul individual
			Prelegeri	Practică/ Seminar	
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	Modalități de evaluare	Termeni de realizare
1. Structuri dinamice de date			
Liste	Portofoliu: Set de aplicații de consolă cu utilizarea tipului de date listă	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 3
Stive și cozi	Portofoliu: Set de aplicații de consolă cu utilizarea tipului de date stive și cozi	Prezentarea portofoliului	Săptămâna 4
Arbori. Arbori binari de căutare	Portofoliu: Set de aplicații de consolă cu utilizarea tipului de date arbori binari	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 6
2. Tehnici de programare			
Analiza algoritmilor	Portofoliu: Set de algoritmi analizați din perspectiva complexității și necesarului de memorie	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 7
Metoda trierii	Portofoliu: Set de aplicații de consolă cu utilizarea metodei trierii	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 8
Metoda Greedy	Portofoliu: Set de aplicații de consolă	Prezentarea portofoliului.	Săptămâna 9

Materii pentru studiul individual	Produse de elaborat	Modalități de evaluare	Termeni de realizare
	cu utilizarea metodei Greedy	Demonstrarea pe calculator	
Metoda reluării	Portofoliu: Set de aplicații de consolă cu utilizarea metodei reluării	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 10
Metoda desparte și stăpânește	Portofoliu: Set de aplicații de consolă cu utilizarea metodei desparte și stăpânește	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 11
Metoda programării dinamice	Portofoliu: Set de aplicații de consolă cu utilizarea metodei programării dinamice	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 12
3. Grafuri			
Reprezentarea grafurilor.	Portofoliu: Set de aplicații de consolă cu utilizarea deferitelor forme de reprezentare a grafurilor.	Prezentarea portofoliului. Demonstrarea pe calculator	Săptămâna 14
Determinare a drumului minim și extragerea arborelui minimal.	Portofoliu: Set de aplicații de consolă de determinare a drumului minim și de extragerea arborelui minimal.	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 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. 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-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.

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.

Problematizarea 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:

- 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 și 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 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. Studiarea 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. Algoritmii de instruire se reprezintă sub forma unui grup de scheme, unui set de operații, iar prin parcurgerea lor într-o ordine bine stabilită duc 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 a competențelor profesionale

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 cunoștința 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.

XI. Resursele necesare pentru desfășurarea procesului de studii

Cerințe față 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
Cerințe 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 2015

XII. Resursele didactice recomandate elevilor

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