



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.01.O.010 Programarea structurată

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.

Pîrvan Evgheni, grad didactic superior, Colegiul „Iulia Hașdeu” din Cahul.

Botoșanu Mihail, grad didactic superior, 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 Incuț 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.....	4
III. Competențele profesionale specifice modulului	5
IV. Administrarea modulului	5
V. Unitățile de învățare	6
VI. Repartizarea orientativă a orelor pe unități de învățare	10
VII. Studiu individual ghidat de profesor.....	10
VIII. Lucrările practice recomandate	11
IX. Sugestii metodologice	11
X. Sugestii de evaluare a competențelor profesionale	13
XI. Resursele necesare pentru desfășurarea procesului de studii	15
XII. Resursele didactice recomandate elevilor	16

I. Preliminarii

La sfârșitul anilor '60 ai secolului XX, datorită dezvoltării vertiginoase a prelucrărilor de date cu calculatorul, s-au putut aborda și rezolva probleme din ce în ce mai complexe. Programele mari, corespunzătoare acestor probleme, s-au complicat în așa măsură încât au devenit foarte greu accesibile chiar și pentru autorii lor. Înțelegerea, testarea, depanarea și modificarea unor astfel de programe prezintă uneori dificultăți de neînlăturat. În acea "criză a software-ului" a apărut necesitatea elaborării unei metodologii generale de realizare în mod sistematic, disciplinat a programelor elegante. Ca răspuns la această aspirație a informaticienilor s-a născut metoda programării structurate.

Statutul Curriculumului. Curriculumul modular "Programarea structurată" 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 specialitățile din domeniul de educație Tehnologia informației și a comunicațiilor;
- membrilor comisiilor pentru examenele de calificare;
- membrilor comisiilor de identificare, evaluare și recunoaștere a rezultatelor învățării, dobândite în contexte non-formale și informale.

Scopul modulului este de transmite elevilor noțiunile fundamentale privind programarea structurată și de a le forma deprinderi practice de lucru în cadrul unui mediu de programare de nivel înalt.

Pentru studierea modulului nu este obligatoriu însușirea anumitor unități de curs.

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 modului

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

CS1. Translarea algoritmilor de decizie/ciclici în cadrul unei aplicații de consolă;

CS2. Elaborarea algoritmilor de rezolvare a problemelor pentru tipuri de date simple/tablou.

IV. Administrarea modului

Semestrul	Numărul de ore			Modalitatea de evaluare	Numărul de credite	
	Total	Contact direct				
		Prelegeri	Practică/ Seminar			
I	120	30	30	60	examen	4

V. Unitățile de învățare

Unități de competență	Unități de conținut	Abilități
1. Elementele de bază ale limbajului de programare		
UC1. Editarea fragmentelor de program.	<ol style="list-style-type: none"> 1. Vocabularul limbajului. Unități lexicale: <ul style="list-style-type: none"> - simbolurile speciale; - cuvintele cheie; - identificatori; - separatori; - numere; - șiruri de caractere. 2. Conceptul de dată. Definierea tipurilor de date: <ul style="list-style-type: none"> - întregi; - reale; - caracter . 3. Operatori. 4. Structura generală a unui program. 5. Variabile, constante. <ul style="list-style-type: none"> - declarare; - inițializare. 6. Operatori. Precedența operatorilor. Conversii. 	<ol style="list-style-type: none"> A1. Scrierea simbolurilor speciale în limbajul de programare. A2. Identificarea cuvintelor cheie ale limbajului de programare. A3. Scrierea identificatorilor. A4. Declararea variabilelor/constantelor conform tipului de dată solicitat. A5. Inițializarea variabilelor/constantelor cu valori prestabilite. A6. Selectarea operatorilor în dependență de specificul tipului de dată utilizat. A7. Scrierea expresiilor aritmetice în limbajul de programare. A8. Identificarea operatorilor și operanzilor. A9. Utilizarea operatorilor și operanzilor la prelucrarea datelor. A10. Utilizarea operatorilor conform specificațiilor propuse. A11. Determinare a compatibilității componentelor instrucțiunilor din punct de vedere al atribuirii. A12. Realizarea conversiilor de date conform specificațiilor propuse. A13. Identificarea părților componente ale unui program.

Unități de competență	Unități de conținut	Abilități
UC2. Elaborarea de programe elementare.	7. Operații de ieșire la nivel de consolă. 8. Operații de intrare la nivel de consolă.	A14. Afișarea valorilor de tip constante. A15. Afișarea valorilor de tip variabilă. A16. Afișarea datelor cu format. A17. Afișarea datelor conform specificațiilor propuse. A18. Citirea datelor conform specificațiilor propuse.
UC3. Scrierea programelor în cadrul mediului de programare.	9. Interfața mediului de programare. 10. Tipuri de instrucțiuni: – vide; – simple; – compuse.	A19. Lansarea mediului de programare. A20. Salvarea unei aplicații de consolă. A21. Deschiderea unei aplicații de consolă. A22. Compilarea unei aplicații de consolă. A23. Executarea unei aplicații de consolă. A24. Personalizarea mediului de programare. A25. Interpretarea rezultatelor unei aplicații de consolă. A26. Testarea unei aplicații de consolă. A27. Evitarea erorilor în cadrul unei aplicații de consolă. A28. Evaluarea expresiilor limbajului de programare. A29. Evaluarea expresiilor propuse și determinarea tipului lor.
2. Structuri de control		
UC4. Utilizarea instrucțiunilor de decizie în cadrul aplicațiilor de consolă.	11. Instrucțiuni de decizie. 12. Instrucțiuni de decizie multiplă.	A30. Identificarea părților componente ale unei instrucțiuni de decizie. A31. Translarea algoritmilor cu decizie unică în limbajul de programare. A32. Translarea algoritmilor cu decizie dublă în limbajul de programare.

Unități de competență	Unități de conținut	Abilități
		<p>A33. Translarea algoritmilor cu decizie triplă în limbajul de programare.</p> <p>A34. Translarea algoritmilor cu decizie multiplă în limbajul de programare.</p> <p>A35. Elaborarea algoritmilor de decizie conform specificațiilor propuse.</p> <p>A36. Implementarea algoritmilor de decizie în limbajul de programare.</p>
UC5. Utilizarea ciclurilor în cadrul aplicațiilor de consolă.	<p>13. Instrucțiunea ciclică for.</p> <p>14. Instrucțiunea ciclică while.</p> <p>15. Particularități ale instrucțiunilor ciclice:</p> <ul style="list-style-type: none"> - întrerupere forțată; - reevaluarea condiției de ciclare. 	<p>A37. Identificarea părților componente ale unei instrucțiuni ciclice.</p> <p>A38. Elaborarea algoritmilor ciclici conform specificațiilor propuse.</p> <p>A39. Elaborarea algoritmilor ciclici cu un număr cunoscut de pași.</p> <p>A40. Elaborarea algoritmilor ciclici cu un număr necunoscut de pași.</p> <p>A41. Elaborarea algoritmilor ciclici cu întrerupere forțată.</p> <p>A42. Elaborarea algoritmilor ciclici cu reevaluarea condiției de ciclare.</p> <p>A43. Translarea algoritmilor ciclici în limbajul de programare.</p> <p>A44. Implementarea algoritmilor ciclici în limbajul de programare.</p>
3. Tipuri de date structurate		
UC6. Prelucrarea tablourilor în cadrul aplicațiilor de consolă.	<p>16. Tablouri unidimensionale.</p> <p>17. Tablouri bidimensionale.</p>	<p>A45. Declararea tablourilor unidimensionale.</p> <p>A46. Declararea tablourilor bidimensionale.</p> <p>A47. Inițializarea elementelor unui tablou</p>

Unități de competență	Unități de conținut	Abilități
		<p>unidimensional/bidimensional.</p> <p>A48. Citirea elementelor tablourilor unidimensionale.</p> <p>A49. Citirea elementelor tablourilor bidimensionale.</p> <p>A50. Afișarea elementelor tablourilor unidimensionale.</p> <p>A51. Afișarea elementelor tablourilor bidimensionale.</p> <p>A52. Prelucrarea elementară a tablourilor unidimensionale.</p> <p>A53. Prelucrarea elementară a tablourilor bidimensionale.</p> <p>A54. Permutarea elementelor tabloului unidimensional.</p> <p>A55. Permutarea elementelor tabloului bidimensional.</p> <p>A56. Eliminarea elementelor din tablou.</p> <p>A57. Interclasarea tablourilor unidimensionale.</p>
UC7. Prelucrarea șirurilor de caractere în cadrul aplicațiilor de consolă.	<p>18. Șiruri de caractere.</p> <p>19. Funcții standard pentru prelucrarea șirurilor de caractere.</p> <p>20. Funcții pentru conversii de date.</p>	<p>A58. Declararea șirurilor de caractere.</p> <p>A59. Utilizarea șirurilor de caractere în cadrul aplicațiilor.</p> <p>A60. Copierea șirurilor de caractere.</p> <p>A61. Concatenarea șirurilor de caractere.</p> <p>A62. Compararea șirurilor de caractere.</p> <p>A63. Căutarea unui șir de caractere.</p> <p>A64. Ștergerea subșirurilor de caractere.</p> <p>A65. Inserarea subșirurilor de caractere.</p> <p>A66. Utilizarea funcțiilor standard la prelucrarea șirurilor de caractere.</p> <p>A67. Conversia datelor din/în șir de caractere.</p>

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.	Elemente de bază ale limbajului de programare	40	10	10	20
2.	Structuri de control	48	12	12	24
3.	Tipuri de date structurate	32	8	8	16
	Total	120	30	30	60

VII. Studiu individual ghidat de profesor

Materii pentru studiul individual	Produce de elaborat	Modalități de evaluare	Termeni de realizare
1. Elemente de bază ale limbajului de programare			
Vocabularul și sintaxa limbajului	Referat. Caracteristici generale ale limbajului de programare	Comunicare	Săptămâna 2
Interfața mediului de programare	Studiu de caz. Interfața mediului de programare	Demonstrare pe calculator	Săptămâna 3
Operații de intrare/ieșire	Portofoliu: Set de aplicații de consolă elementare	Prezentarea portofoliului. Demonstrare pe calculator.	Săptămâna 5
2. Structuri de control			
Instrucțiuni de decizie	Portofoliu: Set de aplicații de consolă cu utilizarea instrucțiunilor de decizie	Prezentarea portofoliului. Demonstrare pe calculator.	Săptămâna 7
Instrucțiuni ciclice	Portofoliu: Set de aplicații de consolă cu utilizarea instrucțiunilor ciclice	Prezentarea portofoliului. Demonstrare pe calculator.	Săptămâna 10
3. Tipuri de date structurate			
Tablouri unidimensionale. Tablouri bidimensionale.	Studiu de caz: Implementarea	Comunicare	Săptămâna 11

Materii pentru studiul individual	Produse de elaborat	Modalități de evaluare	Termeni de realizare
	algoritmului de căutare binară.		
	Portofoliu: Set de aplicații de consolă pentru prelucrarea tablourilor.	Prezentarea portofoliului. Demonstrare pe calculator.	Săptămâna 13
Șiruri de caractere.	Portofoliu: Set de aplicații de consolă pentru prelucrarea șirurilor de caractere.	Prezentarea portofoliului. Demonstrare pe calculator.	Săptămâna 15

VIII. Lucrările practice recomandate

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

1. Introducerea și extragerea datelor la nivel de consolă.
2. Evaluarea expresiilor aritmetice.
3. Implementarea algoritmilor de decizie.
4. Implementarea algoritmilor ciclici.
5. Prelucrarea tablourilor unidimensionale.
6. Prelucrarea tablourilor bidimensionale.
7. Prelucrarea șirurilor de caractere.

IX. Sugestii metodologice

Elementul de bază al Curriculumului sunt competențele ce trebuie formate și dezvoltate în procesul de formare profesională. Acestea vor fi formate 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:

- Structura generală a unui program;
- Conceptul de dată. Definirea tipurilor de date;
- Tipuri de instrucțiuni;
- Instrucțiuni ciclice.

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 situației 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 a 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:

- Instrucțiuni de decizie;
- Instrucțiuni de decizie multiplă;
- Instrucțiuni ciclice;
- Tablouri.

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:

- Instrucțiuni de decizie;
- Instrucțiuni ciclice;
- Tablouri.

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:

- Interfața mediului de programare;
- Operații de intrare/ieșire la nivel de consolă.

precum și în cadrul orelor de laborator.

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 structurată*, 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 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 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;
- 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.

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.	E. Cerchez, M. Șerban, Programarea în limbajul C/C++ pentru liceu, 2005. http://cnvranceanu.ro/blog/wp-content/uploads/2014/11/cerchez-verde-m.pdf	Internet
2.	G. Vasilache, S. Gîncu Culegere de probleme la informatica, Chișinău, 2012. http://en.calameo.com/read/002801569a611d413be1c	Internet
3.	B. Eckel, Thinking in C++, 2nd Edition, Vol 1 and Vol 2. http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html	Internet
4.	A. Ruceanu, Curs C++. http://www.ruceanu.ro/adrian/wp-content/cursuri/S1progr2013.php	Internet
5.	Manual de Programare C++. https://ro.scribd.com/doc/15449644/Manual-de-Programare-C	Internet
6.	Cărți pentru programatori. http://www.infoarena.ro/blog/carti	Internet

Tutoriale pe Internet

1. <http://www.programming.com>
2. <http://www.cplusplus.com>
3. <http://www.infoarena.ro/>