



**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.02.O.012 Programarea procedurală**

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 „Lulia Hașdeu” din Cahul.

**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 .....	10
VII. Studiu individual ghidat de profesor .....	10
VIII. Lucrările practice recomandate .....	11
IX. Sugestii metodologice .....	11
X. Sugestii de evaluare a competențelor profesionale .....	11
XI. Resursele necesare pentru desfășurarea procesului de studii .....	16
XII. Resursele didactice recomandate elevilor .....	16

## I. Preliminarii

Programarea procedurală este o paradigmă de programare bazată pe conceptul de apel de procedură (subprogram). Procedurile, numite și rutine, subrutine, metode sau funcții conțin o serie de pași computaționali care trebuie parcurși. Orice procedură poate fi apelată oriunde în timpul execuției unui program inclusiv din interiorul ei (apel recursiv).

*Statutul Curriculumului.* Curriculumul modular “Programarea procedurală” 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 de date structurate și a subprogramelor în mentenanța și actualizarea produselor-program (softurilor) de sistem precum și a altor 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 unități de curs:

- Programarea structurată.

## 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 de date structurate în cadrul aplicațiilor de consolă;
- CS2. Organizarea programelor la nivel de subprograme;
- CS3. Elaborarea de programare pentru problemele întâlnite în activitatea profesională.

## IV. Administrarea modulului

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

## V. Unitățile de învățare

Unități de competență	Unități de conținut	Abilități
<b>1. Tipuri de date structurate</b>		
UC1. Utilizarea fișierelor în cadrul aplicațiilor de consolă	1. Tipul de date fișier. Operații de specifice tipului de date fișier: <ul style="list-style-type: none"> <li>– citire;</li> <li>– scriere;</li> <li>– funcții de prelucrare a fișierelor.</li> </ul>	A1. Declararea variabilelor de tip fișier. A2. Deschiderea fișierelor conform specificațiilor propuse. A3. Închiderea fișierelor. A4. Verificarea sfârșitului de fișier. A5. Citirea datelor din fișiere. A6. Scrierea datelor în fișiere. A7. Utilizarea funcțiilor de prelucrare a fișierelor conform specificațiilor propuse.
UC2. Sortarea datelor structurate în cadrul aplicațiilor de consolă	2. Metode de sortare: <ul style="list-style-type: none"> <li>– metoda de inserție;</li> <li>– metoda bulelor;</li> <li>– metoda de selecție;</li> <li>– sortare prin distribuire.</li> </ul>	A8. Selectarea algoritmilor de sortare a datelor structurate. A9. Prezentarea situațiilor de aplicare a metodelor de sortare. A10. Detalierea algoritmilor de sortare a datelor structurate. A11. Translarea algoritmilor de sortare în limbajul de programare. A12. Implementarea algoritmilor de sortare în limbajul de programare.
UC3. Prelucrarea tipurilor de date de tip structură în cadrul	3. Structuri. 4. Tablouri de structuri.	A13. Declararea structurilor de date conform specificațiilor propuse.

Unități de competență	Unități de conținut	Abilități
aplicațiilor de consolă		<p>A14. Declararea variabilelor de tip structură.</p> <p>A15. Declararea tablourilor de tip structură.</p> <p>A16. Inițializarea variabilelor de tip structură.</p> <p>A17. Accesarea câmpurilor unei structuri.</p> <p>A18. Citirea/afișarea datelor unei tip de date structură.</p> <p>A19. Afișarea structurilor.</p> <p>A20. Implementarea operațiilor asupra structurilor conform specificațiilor propuse.</p> <p>A21. Sortarea tablourilor de structuri.</p> <p>A22. Utilizarea tablourilor de structuri în cadrul aplicațiilor de consolă.</p>
UC4. Utilizarea pointerilor în cadrul aplicațiilor de consolă	<p>5. Pointer. Variabile pointer.</p> <p>6. Alocarea dinamică memoriei. Crearea dinamică a tipului de date tablou.</p> <p>7. Relații dintre pointeri și diverse tipuri de date.</p>	<p>A23. Declararea unei variabile de tip pointer.</p> <p>A24. Utilizarea operatorului de adresare.</p> <p>A25. Utilizarea operatorului de diferențiere.</p> <p>A26. Setarea legăturilor pointer-tablou.</p> <p>A27. Accesarea datelor tablourilor unidimensionale/structurilor prin intermediul pointerului.</p> <p>A28. Aplicarea operațiilor asupra pointerilor conform specificațiilor propuse.</p> <p>A29. Alocarea dinamică a memoriei pentru tipurile de date numerice/caracter.</p> <p>A30. Alocarea dinamică a memoriei pentru tablouri.</p>

Unități de competență	Unități de conținut	Abilități
		A31. Eliberarea memoriei.
<b>2. Subprograme</b>		
UC5. Utilizarea subprogramelor în cadrul aplicațiilor de consolă	<p>8. Noțiuni de subprograme.</p> <ul style="list-style-type: none"> <li>– subprograme cu tip;</li> <li>– subprograme fără tip.</li> </ul> <p>9. Domenii de vizibilitate:</p> <ul style="list-style-type: none"> <li>– structura de bloc a programelor;</li> <li>– variabile globale și variabile locale.</li> </ul> <p>10. Efecte colaterale:</p> <ul style="list-style-type: none"> <li>– cauzele efectelor colaterale;</li> <li>– modul de evitare a efectelor colaterale.</li> </ul>	<p>A32. Utilizarea terminologiei specifice subprogramelor.</p> <p>A33. Declararea subprogramelor cu/fără tip.</p> <p>A34. Declararea anticipată a subprogramelor.</p> <p>A35. Apelarea subprogramelor cu/fără tip.</p> <p>A36. Elaborarea subprogramelor cu tip/fără tip.</p> <p>A37. Elaborarea subprogramelor cu parametri de tip valoare/adresă.</p> <p>A38. Prelucrarea datelor utilizate în procesul de studiu la nivel de subprograme.</p> <p>A39. Comunicarea între subprograme prin intermediul variabilelor globale/parametrilor.</p> <p>A40. Evitarea efectelor colaterale.</p> <p>A41. Utilizarea subprogramelor definite de către utilizator în cadrul programelor.</p> <p>A42. Prelucrarea datelor structurate la nivel de subprograme.</p> <p>A43. Elaborarea algoritmilor la nivel de subprograme.</p> <p>A44. Implementarea algoritmilor bazați pe subprograme în limbajul de programare.</p>
UC6. Utilizarea subprogramelor recursive în cadrul aplicațiilor	11. Subprograme recursive.	A45. Descrierea subprogramelor recursive.



Unități de competență	Unități de conținut	Abilități
de consolă		<p>A46. Prezentarea situațiilor de aplicare a subprogramelor recursive.</p> <p>A47. Elaborarea subprogramelor recursive cu cel puțin doi parametri.</p> <p>A48. Elaborarea subprogramelor recursive indirecte.</p> <p>A49. Apelarea subprogramelor recursive.</p> <p>A50. Evitarea cazurilor de supraîncărcare a stivei.</p> <p>A51. Elaborarea algoritmilor recursivi pentru prelucrarea datelor utilizate în procesul de studiu.</p> <p>A52. Implementarea algoritmilor recursivi în limbajul de programare.</p>
UC7. Organizarea aplicațiilor de consolă la nivel de module	<p>12. Programarea modulară:</p> <ul style="list-style-type: none"> <li>– unit-uri (module);</li> <li>– supraîncărcare.</li> </ul>	<p>A53. Crearea unit-urilor.</p> <p>A54. Supraîncărcarea subprogramelor.</p> <p>A55. Utilizarea unit-urilor definite de către utilizator în cadrul aplicațiilor de consolă.</p> <p>A56. Elaborarea aplicațiilor de consolă bazate pe unit-uri.</p> <p>A57. Elaborarea proiectelor modulare conform specificațiilor propuse.</p> <p>A58. Gestiunea datelor tablourilor de tip structură la nivel de module.</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.	Tipuri de date structurate	64	16	16	32
2.	Subprograme	56	14	14	28
	<b>Total</b>	<b>120</b>	<b>30</b>	<b>30</b>	<b>60</b>

## VII. Studiu individual ghidat de profesor

Materii pentru studiul individual	Produse de elaborat	Modalități de evaluare	Termeni de realizare
<b>1. Tipuri de date structurate</b>			
Fișiere.	Portofoliu: Set de aplicații de consolă cu utilizarea tipului de date fișier	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 2
Metode de sortare.	Portofoliu: Set de aplicații de consolă pentru sortarea datelor prin diverse metode.	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 4
Structuri.	Portofoliu: Set de aplicații de consolă pentru prelucrarea structurilor.	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 7
Pointeri	Portofoliu: Set de aplicații de consolă ce utilizează variabile de tip pointer.	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 8
<b>2. Subprograme</b>			
Elaborarea de subprograme	Portofoliu: Set de aplicații de consolă ce utilizează subprograme	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 12

<b>Materii pentru studiul individual</b>	<b>Produse de elaborat</b>	<b>Modalități de evaluare</b>	<b>Termeni de realizare</b>
Subprograme recursive	Portofoliu: Set de aplicații de consolă ce utilizează subprograme recursive	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 14
Programare modulară	Portofoliu: Set de aplicații de consolă ce structurate pe module	Prezentarea portofoliului. Demonstrarea pe calculator.	Săptămâna 15

### **VIII. Lucrările practice recomandate**

1. Citirea și scrierea datelor din/în fișiere.
2. Sortarea datelor.
3. Procesarea structurilor.
4. Pointeri.
5. Subprograme cu tip.
6. Subprograme fără tip.
7. Subprograme recursive.
8. Module.

### **IX. Sugestii metodologice**

Elementul de bază al Curriculumului sunt competențele ce trebuie formate și dezvoltate în procesul de instruire 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:

- Metode de sortare;
- Pointer. Alocarea dinamică a memoriei;
- Subprograme recursive.

**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:

- Tipul de date fișier;
- Structuri;
- Efecte colaterale;
- Supraîncărcarea funcțiilor.

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

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

- Metode de sortare;
- Tablouri de structuri;
- Subprograme recursive;
- Programare modulară.

**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:

- Metode de sortare;
- Subprograme recursive;
- Programare modulară.

**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:

- Tipul de date fișier. Operații de specifice tipului de date fișier.
- Metode de sortare;
- Structuri;
- Subprograme.

**Instruirea prin proiecte** reprezintă o modalitate de instruire/autoinstruire grație căreia elevii, dar mai ales elevii efectuează o cercetare orientată spre obiective practice și finalizată într-un produs ce poate fi un obiect, un aparat, o instalație, o culegere tematică, un album, o lucrare științifică etc. Această metodă se recomandă pentru predarea-învățarea-evaluarea următoarelor unități de conținut:

- Metode de sortare;
- Tablouri de structuri;
- Subprograme;
- Programarea modulară.

## 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 procedurală*, 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 de date conform specificațiilor tehnice;
- elaborarea algoritmilor pentru tipurile de date utilizate în procesul de studiu;
- 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;
- subprograme elaborate conform specificațiilor propuse;
- algoritmi elaborați conform specificațiilor propuse;

- structuri 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

<b>Cerințe față de sălile de curs</b>	
<b>Pentru orele teoretice</b>	Cabinet de informatică cu 15 calculatoare Proiector
<b>Pentru orele de laborator</b>	Laborator de informatică care asigură fiecărui elev un calculator
<b>Cerințe tehnice</b>	
<b>Parametri tehnici minimi ale calculatorului</b>	<b>Procesor:</b> 2 GHz <b>Memorie operativă:</b> 4 GB <b>Unitate de stocare:</b> 500 GB <b>Afișaj și grafică:</b> size: 22", resolution: 1366x768 <b>Network:</b> Ethernet, 100 Mb
<b>Software</b>	Sistem de Operare Microsoft Windows Code::Blocks Dev C/CPP Visual Studio 2015

#### XII. Resursele didactice recomandate elevilor

<b>Nr. crt.</b>	<b>Denumirea resursei</b>	<b>Locul în care poate fi consultată/ accesată resursa</b>
1.	A. Ruceanu, Proiectarea algoritmilor. <a href="http://www.ruceanu.ro/adrian/wp-content/cursuri/pa2014.php">http://www.ruceanu.ro/adrian/wp-content/cursuri/pa2014.php</a>	Internet
2.	A. Deaconu, Programare procedurală, 2010. <a href="http://server1.docfoc.com/uploads/Z2016/01/06/P9U8EXFUZR/bf5627d4107759da24273d945bcacb16.pdf">http://server1.docfoc.com/uploads/Z2016/01/06/P9U8EXFUZR/bf5627d4107759da24273d945bcacb16.pdf</a>	Internet
3.	E. Cerchez, M. Șerban, Programarea în limbajul C/C++ pentru	Internet



	liceu, 2005. <a href="http://en.calameo.com/read/002801569a611d413be1c">http://en.calameo.com/read/002801569a611d413be1c</a>	
4.	G. Vasilache, S. Gîncu Culegere de probleme la informatica, Chişinău, 2012. <a href="http://en.calameo.com/read/002801569a611d413be1c">http://en.calameo.com/read/002801569a611d413be1c</a>	Internet
5.	H. Oros, Programare procedurală. <a href="http://webhost.uoradea.ro/horos/files/PP.pdf">http://webhost.uoradea.ro/horos/files/PP.pdf</a>	Internet
6.	Elemente de algoritmică și limbaje de programare. <a href="https://ro.scribd.com/document/48218378/21297279-programare-procedurala">https://ro.scribd.com/document/48218378/21297279-programare-procedurala</a>	Internet

### **Tutoriale pe Internet**

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