



Ministerul Educației și Cercetării al Republicii Moldova

I.P. Centrul de Excelență în Energetică și Electronică



Mariana BARLADEAN,
Directoare I.P. Centrul de Excelență în
Energetică și Electronică

15 septembrie 2025

Curriculum la modulul

S.08.O.020 Limbaje de programare

Specialitatea: **0611.1 Calculatoare**

Calificarea: **0611.1.1 Tehnician/tehniciană suport tehnic al
calculatoarelor**

Chișinău 2025

Curriculumul a fost elaborat în baza ordinului Ministerului Educației și Cercetării al Republicii Moldova, numărul 1229/2024, cu privire la aprobarea listei instituțiilor de învățământ desemnate responsabile pentru elaborarea planurilor de învățământ la programele de formare profesională postsecundară și postsecundar nonterțială.



Aprobat la:

Consiliul metodic-științific al I.P. Centrul de Excelență în Energetică și Electronică din
15 septembrie 2025, directoare Mariana BARLADEAN



Autor:

- 1) Lucia SINEAVSCHI, profesor discipline de specialitate, grad didactic superior, șef secție studii, I.P. CEEE

Recenzenți:

1. GAMA COMPUTER SRL, adresa: str. Meșterul Manole 12/2, mun. Chișinău, Director: Mincheivici Sergiu;
2. S.R.L. RC-ELECTRON, str. Calea Ieșilor, 16/1, mun. Chișinău, Director: Cebanu Radu.

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.....	6
V. Unitățile de învățare	6
VI. Repartizarea orientativă a orelor pe unități de învățare.....	8
VII. Studiu individual ghidat de profesor	9
VIII. Lucrările practice recomandate	10
IX. Sugestii metodologice	10
X. Sugestii de evaluare a rezultatelor învățării	12
XI. Resursele necesare pentru desfășurarea procesului de studiu.....	12
XII. Resursele didactice recomandate elevilor	13

I. Preliminarii

Curriculumul la modulul **Limbaje de programare** este elaborat în baza ordinului MEC 768/2024 *Cu privire la aprobarea Referențialului de corelare a programelor de studii (specialităților) și calificărilor pentru învățământul profesional tehnic postsecundar și postsecundar nonterțiar și a planului de învățământ* ediția 2024, aprobat prin Ordinul MEC nr. 1229/2024. Modulul **Limbaje de programare** este un element formativ al pachetului curricular pentru calificările din domeniul de formare profesională *Utilizarea calculatorului* și face parte din componenta de specialitate a planului de învățământ la programul de studii 0611.1 Calculatoare.

Scopul modulului constă în formarea la elevi a competențe fundamentale în gândirea algoritmică, analiza și rezolvarea problemelor prin mijloace informatice, precum și în dezvoltarea de aplicații software utilizând diferite paradigme și limbaje de programare, necesare pentru pregătirea corespunzătoare în specialitate a unui absolvent al IPT.

Pentru demararea procesului instructiv la modulului **Limbaje de programare** sunt necesare cunoștințe la disciplina Informatică, Matematică, Fizică.

II. Motivația, utilitatea modulului pentru dezvoltarea profesională

Curriculumul la modulul **Limbaje de programare** reprezintă documentul normativ de bază care descrie condițiile învățării și performanțele ce trebuie atinse la disciplină, exprimate în rezultate ale învățării.

Prin studierea modulului **Limbaje de programare** se urmărește formarea la elevi a următoarelor valori și atitudini:

- Rigoare, precizie și responsabilitate în elaborarea și testarea programelor;
- Perseverență și spirit analitic în identificarea și corectarea erorilor;
- Creativitate și inițiativă în proiectarea soluțiilor informatice și în dezvoltarea aplicațiilor;
- Gândire logică și sistemică, necesară pentru înțelegerea și organizarea proceselor informatice complexe;
- Autonomie în învățare și dorință de perfecționare continuă, având în vedere evoluția rapidă a tehnologiilor;
- Spirit de colaborare și muncă în echipă, specific mediului profesional din domeniul IT;
- Respect față de munca proprie și a altora, prin respectarea normelor de etică profesională și a drepturilor de autor;
- Atitudine responsabilă față de utilizarea resurselor digitale, promovând securitatea informațională și utilizarea corectă a tehnologiilor;
- Deschidere spre inovare și utilizarea tehnologiei în scopul progresului personal și social.

Limbaje de programare este modulul indispensabil oricărui specialist în domeniul calculatoare și tehnologia informației, care contribuie la formarea competențelor profesionale de analiză, proiectare, implementare și testare a soluțiilor algoritmice pentru diverse probleme tehnice și aplicative.

Studierea modulului **Limbaje de programare** asigură dobândirea de către elevi a cunoștințelor, abilităților și deprinderilor necesare pentru:

- Elaborarea, testarea și documentarea programelor;
- Citirea și interpretarea codului sursă;

- Aplicarea corectă a conceptelor fundamentale ale programării, precum variabile, constante, operatori, structuri de control, funcții, tablouri, șiruri de caractere, structuri și pointeri;
- Manipularea datelor prin fișiere text și binare;
- Implementarea algoritmilor modulari și recursivi;
- Aplicarea principiilor programării structurate și modulare în rezolvarea problemelor practice.

Conținutul modului Limbaje de programare este axat pe dezvoltarea gândirii algoritmice, capacității de analiză și sinteză, precum și pe formarea competențelor practice necesare pentru realizarea programelor funcționale și eficiente, ceea ce va asigura formarea abilităților de proiectare, codare, testare și optimizare a aplicațiilor software.

Modulul *Limbaje de programare* creează condițiile necesare pentru studierea următoarelor unități de curs:

S.08.O.023 Securitatea informațională

S.08.A.030 Sisteme electronice programabile

S.08.L.034 Robotica

III. Competențele profesionale specifice modului

Modulul *Limbaje de programare* va contribui la formarea rezultatelor învățării descrise în Standardul de calificare Tehnician/tehniciană suport tehnic al calculatoarelor care, la rândul lor, vor asigura formarea competențelor profesionale specificate în același document:

Rezultate ale învățării	Competențe profesionale specifice
RÎ2. Absolventul elaborează, testează și depanează programe utilizând un limbaj de programare de nivel înalt. RÎ3. Absolventul utilizează funcții, proceduri și structuri de control pentru realizarea aplicațiilor informatice adaptate cerințelor practice. RÎ4. Absolventul documentează corect codul sursă, testele efectuate și rezultatele obținute conform cerințelor tehnice. RÎ6. Absolventul colaborează eficient în echipă pentru dezvoltarea, testarea și implementarea aplicațiilor software. RÎ5. Absolventul aplică standarde și bune practici de programare, respectând normele de securitate informatică și proprietate intelectuală.	CP2 – Testarea componentelor hardware CP3 – Implementarea soluțiilor CP4 – Elaborarea documentației CP5 – Asistența tehnică a utilizatorilor CP11 – Managementul riscurilor

Pentru a atinge rezultatul învățării în Standardul de calificare Tehnician/tehniciană suport tehnic al calculatoarelor, după studierea modului *Limbaje de programare*, elevul va fi capabil:

- să identifice și să utilizeze corect conceptele fundamentale ale limbajelor de programare;
- să scrie, să testeze și să depaneze programe simple în limbajul predat;
- să aplice funcții și structuri de control în rezolvarea unor probleme practice;
- să utilizeze funcții standard pentru prelucrarea șirurilor de caractere și a altor tipuri de date;

- să elaboreze aplicații informatice simple care să răspundă unor cerințe date;
- să demonstreze o atitudine responsabilă și organizată în proiectarea și implementarea soluțiilor software;
- să lucreze individual și în echipă, respectând cerințele proiectului și standardele profesionale.

IV. Administrarea modului

Semestrul	Numărul de ore					Forma de evaluare	Numărul de credite
	Total	Contact direct			Studiu individual		
		Teorie	Practică	Laborator			
VIII	90	40	-	20	30	examen	3

V. Unitățile de învățare

Rezultatele învățării	Unități de conținut	Abilități
1. Noțiuni introductive. Structura unui program.		
RÎ1. Utilizarea elementelor de bază ale limbajului de programare	1.1 Rolul limbajelor de programare în formarea gândirii algoritmice. Clasificarea limbajelor de programare. Etapele realizării unui program. Mediile de programare. 1.2 Elemente de bază ale unui program .Tipuri de date și variabile. Constante și operatori. Instrucțiuni de intrare/ieșire.	1. Identificarea și utilizarea corectă a elementelor de bază ale unui limbaj de programare. 2. Aplicarea etapelor de dezvoltare a unui program simplu. 3. Folosirea unui mediu de programare pentru a scrie, salva, executa și verifica programe simple. 4. Explicarea rolului limbajelor de programare în dezvoltarea gândirii logice și algoritmice.
2. Structuri de control		
RÎ2. Aplicarea structurilor de control	2.1 Instrucțiuni secvențiale. Instrucțiuni alternative (if, if...else, switch). 2.2 Instrucțiuni repetitive (for, while, do...while). 2.3 Construcții algoritmice uzuale (căutare, sortare, selecție)	5. Identificarea și utilizarea corectă a structurilor de control. 6. Analiza și selectarea tipul potrivit de structură de control în funcție de logica problemei de rezolvat. 7. Elaborarea algoritmilor care integrează structuri de control pentru realizarea unor prelucrări uzuale. 8. Testarea, interpretarea și corectarea erorile din programele care conțin structuri

Rezultatele învățării	Unități de conținut	Abilități
		de control, demonstrând înțelegerea funcționării acestora.
3. Funcții și proceduri		
RÎ3. Elaborarea programelor modulare	3.1 Noțiunea de subprogram. Declararea și apelarea funcțiilor. 3.2 Transmiterea parametrilor prin valoare și prin referință 3.3 Recursivitatea. Modularizarea programului.	9. Definirea și utilizarea funcțiilor proprii pentru structurarea logică a programului. 10. Aplicarea corectă a mecanismelor de transmitere a parametrilor prin valoare și prin referință, în funcție de scopul subprogramului. 11. Implementarea algoritmilor recursivi pentru rezolvarea problemelor care implică autoapelul funcției. 12. Elaborarea programelor modulare, structurate pe funcții și proceduri interdependente, pentru simplificarea întreținerii și extinderea codului.
4. Structuri de date		
RÎ4. Manipularea structurilor de date	4.1. Tablouri unidimensionale și bidimensionale. Șiruri de caractere. 4.2. Structuri și înregistrări. 4.3. Utilizarea pointerilor.	13. Definirea și utilizarea tablourilor unidimensionale și bidimensionale pentru stocarea și prelucrarea datelor. 14. Gestionarea șirurilor de caractere prin operații de citire, comparare, concatenare și prelucrare. 15. Utilizarea structurilor și înregistrărilor pentru gruparea logică a datelor complexe. 16. Realizează prelucrări complexe asupra datelor stocate în structuri și tablouri prin combinarea instrucțiunilor iterative și condiționale.

Rezultatele învățării	Unități de conținut	Abilități
		17. Aplicarea pointerilor pentru accesarea și manipularea eficientă a datelor în tablouri, structuri sau șiruri de caractere. 18. Dezvoltă programe modulare care integrează tablouri, structuri și pointeri pentru optimizarea codului și reutilizarea funcțiilor.
5. Prelucrarea fișierelor		
RÎ5. Prelucrarea datelor din fișiere	5.1 Tipuri de fișiere (text, binare). Deschiderea, citirea, scrierea și închiderea fișierelor. 5.2 Salvarea și prelucrarea datelor din fișiere. 5.3 Aplicarea structurilor de control în prelucrarea datelor din fișiere.	19. Deschiderea, citirea, scrierea și închiderea fișiere corect, respectând regulile limbajului de programare. 20. Aplicarea structurilor de control (secvențiale, alternative, repetitive) în prelucrarea datelor din fișiere. 21. Elaborarea algoritmilor modulari pentru prelucrarea fișierelor, folosind funcții pentru citire, procesare și scriere. 22. Testarea și corectarea erorilor în programele care manipulează fișiere, asigurând integritatea datelor.

VI. Repartizarea orientativă a orelor pe unități de învățare

Nr. crt.	Unități de învățare	Numărul de ore			
		Total	Contact direct		Studiu individual
			Teorie	Practică	
1.	Noțiuni introductive. Structura unui program.	8	4	2	2
2.	Structuri de control	18	8	4	6
3.	Funcții și proceduri	18	8	4	6
4.	Structuri de date	24	10	6	8
5.	Prelucrarea fișierelor	22	10	4	8
	Total	90	40	20	30

VII. Studiu individual ghidat de profesor

Materii pentru studiul individual	Produse de elaborat	Modalități de evaluare
Noțiuni introductive. Structura unui program.		
1.1 Rolul limbajelor de programare în formarea gândirii algoritmice. Clasificarea limbajelor de programare. Etapele realizării unui program. Mediile de programare. 1.2 Elemente de bază ale unui program .Tipuri de date și variabile. Constante și operatori. Instrucțiuni de intrare/ieșire.	investigație	Prezentarea rezultatelor cercetării și a concluziilor, utilizând desene, grafice, diagrame, tabele
Structuri de control		
2.1 Instrucțiuni secvențiale. Instrucțiuni alternative (if, if...else, switch). 2.2 Instrucțiuni repetitive (for, while, do...while). 2.3 Construcții algoritmice uzuale (căutare, sortare, selecție)	proiect	Prezentarea proiectului
Funcții și proceduri		
3.1 Noțiunea de subprogram. Declararea și apelarea funcțiilor. 3.2 Transmiterea parametrilor prin valoare și prin referință 3.3 Recursivitatea. Modularizarea programului.	Ghid / fișă explicativă despre funcții	Prezentarea rezultatelor
Structuri de date		
4.1. Tablouri unidimensionale și bidimensionale. Șiruri de caractere. 4.2. Structuri și înregistrări. 4.3. Utilizarea pointerilor.	Proiect complex aplicativ	Susținerea proiectului
Prelucrarea fișierelor		
5.1 Tipuri de fișiere (text, binare). Deschiderea, citirea, scrierea și închiderea fișierelor. 5.2 Salvarea și prelucrarea datelor din fișiere. 5.3 Aplicarea structurilor de control în prelucrarea datelor din fișiere.	Ghid practic de lucru cu fișiere	Prezentarea ghidului

Notă: Termenele de executare a studiului individual ghidat de profesor vor fi specificate în *Planul de lungă durată*.

VIII. Lucrările de laborator recomandate

Unități de învățare	Lista lucrărilor de laborator	Ore
Noțiuni introductive. Structura unui program.	1. Analiza mediului de programare și etapele dezvoltării unui program informatic care utilizează variabile, constante și operații elementare.	2
Structuri de control	2. Analiza și implementarea structurilor secvențiale și alternative în limbajul de programare C/C++. 3. Implementarea instrucțiunilor repetitive for, while și do...while în limbajul de programare C/C++.	4
Funcții și proceduri	4. Implementarea subprogramelor și transmiterea parametrilor în limbajul C/C++. 5. Implementarea recursivității și modularizarea programelor în limbajul C/C++.	4
Structuri de date	6. Manipularea tablourilor și șirurilor de caractere în limbajul C/C++. 7. Utilizarea structurilor și înregistrărilor în limbajul C/C++. 8. Utilizarea pointerilor în limbajul C/C++.	6
Prelucrarea fișierelor	9. Manipularea fișierelor text și binare în limbajul C/C++. 10. Aplicarea structurilor de control în prelucrarea datelor din fișiere în limbajul C/C++.	4
Total		20

IX. Sugestii metodologice

Conținuturile modului ***Limbaje de programare*** trebuie să fie abordate într-o manieră flexibilă, diferențiată, ținând cont de particularitățile elevilor, de stilurile individuale de învățare, precum și de nivelul inițial de pregătire al acestora, în mod special în domeniul informaticii, matematicii.

Studiul conținuturilor și efectuarea activităților practice se fac în ordinea propusă în coloana „Unități de conținut”, dar numărul de ore alocat fiecărei teme și activități practice rămâne la decizia cadrelor didactice care predau disciplina. Aceștia vor repartiza timpul în funcție de gradul de dificultate al temelor, dar și de ritmul de asimilare a cunoștințelor și de formare a abilităților de către elevi.

Strategii didactice

Pentru a atinge rezultate ale învățării care să contribuie la dezvoltarea competențelor profesionale, strategiile didactice la modulul *Limbaje de programare* vor fi axate preponderent pe activizarea structurilor funcțional-acționare ale elevilor.

Dat fiind caracterul practic al procesului de instruire la modulul *Limbaje de programare*, profesorul va selecta metode și tehnici didactice axate pe formarea aptitudinilor și deprinderilor praxiologice necesare. Pentru a atinge rezultatele învățării scontate, se recomandă aplicarea metodelor și tehnologiilor de predare care fac parte din:

- instruirea problematizată;
- instruirea euristică;
- instruirea demonstrativă.

În acest sens, sunt binevenite metodele algoritmice, bazate pe acțiune reală și autentică, precum: algoritmizarea, exercițiul practic, utilizarea fișelor de lucru și studii de caz. Un rol central în formarea deprinderilor practice îl ocupă lucrările de laborator, axate pe dezvoltarea următoarelor abilități:

- analiza și proiectarea algoritmilor pentru rezolvarea problemelor specifice limbajelor de programare C/C++;
- implementarea corectă a structurilor de control (secvențiale, alternative și repetitive) în programe funcționale;
- utilizarea corectă a funcțiilor și subprogramelor, inclusiv transmiterea parametrilor prin valoare și referință, precum și aplicarea recursivității;
- prelucrarea datelor complexe prin tablouri unidimensionale și bidimensionale, șiruri de caractere, structuri și înregistrări;
- manipularea fișierelor text și binare, inclusiv citirea, scrierea și gestionarea datelor persistente;
- aplicarea pointerilor pentru manipularea eficientă a memoriei și transmiterea datelor prin referință;
- testarea și verificarea corectitudinii programelor, interpretarea rezultatelor și documentarea clară a algoritmilor și codului sursă;
- dezvoltarea gândirii algoritmice și a independenței în învățare, prin realizarea portofoliului de programe și a temelor practice.

Pentru formarea gândirii logico-creative, se recomandă aplicarea metodelor didactice axate pe:

- rezolvarea de probleme și exerciții aplicate – dezvoltarea capacității de analiză, sinteză și aplicare a conceptelor teoretice în situații concrete;
- algoritmizarea și proiectarea programelor – stimularea gândirii structurate și a planificării logice a soluțiilor;
- lucrul cu fișe de lucru și studii de caz – consolidarea deprinderilor prin exemple practice și sarcini autentice;
- dezvoltarea creativității în programare – încurajarea generării de soluții originale, optimizarea algoritmilor și experimentarea cu variante alternative;
- învățarea prin descoperire și explorare – încurajarea investigării independente a limbajului de programare și a funcționalităților sale;
- colaborarea și activități de grup – schimbul de idei, rezolvarea problemelor în echipă și discutarea strategiilor algoritmice;
- autoevaluarea și reflecția asupra muncii proprii – evaluarea progresului individual prin portofoliu, teste practice și analiza rezultatelor obținute.

În procesul de predare-învățare, profesorul va pune accentul pe învățământul formativ-dezvoltativ, individual și de grup.

X. Sugestii de evaluare a rezultatelor învățării

Evaluarea este implicită demersului didactic și urmărește măsura în care au fost atinse rezultatele învățării exprimate prin cunoștințe și deprinderi/abilități. Evaluarea permite atât profesorului, cât și elevului să cunoască nivelul deprinderilor și cunoștințelor dobândite, să identifice lacunele și cauzele lor, fapt care oferă un feedback eficient pentru reglarea procesului de predare-învățare. Prin urmare, în cadrul demersului evaluativ vor fi valorificate diverse metode, tehnici și instrumente de evaluare care contribuie la optimizarea procesului de învățare.

Evaluarea continuă (formativă) a elevilor va fi efectuată de către cadrele didactice pe baza unor probe explicite, corespunzătoare rezultatelor învățării scontate. Metode de evaluare formativă recomandate sunt:

- *observarea sistematică* a comportamentului elevilor, care permite evaluarea înțelegerii conceptelor, a proceselor, a procedurilor, a aptitudinilor, dar și a atitudinilor față de o sarcină dată;
- *autoevaluarea*, prin care elevul compară nivelul la care a ajuns cu obiectivele și standardele educaționale și își poate modifica programul propriu de învățare;
- *evaluarea reciprocă*, metodă care contribuie la responsabilizarea, conștientizarea, participarea elevilor la evaluare, la socializarea grupului de elevi, la coparticiparea acestora la formarea semenilor, precum și la intensificarea convingerilor proprii.

În calitate de *instrumente de evaluare* pot fi folosite:

- fișe de observație;
- fișe de autoevaluare;
- fișe de evaluare.

Evaluarea sumativă la modulul *Limbaje de programare* va măsura gradul de atingere a rezultatelor învățării și va fi desfășurată sub formă de examen, cu două etape:

1. partea teoretică: test docimologic, axat preponderent pe evaluarea domeniului cognitiv;
2. partea practică: sarcini practice.

Pentru a fi admis la examen, elevul trebuie să:

- prezinte portofoliul studiului individual, care reflectă progresul și experiența dobândite în învățarea independentă;
- realizeze și să prezinte lucrările de laborator, care permit evaluarea cunoștințelor teoretice și a competențelor practice în aplicarea corectă a conceptelor și algoritmilor studiați.

XI. Resursele necesare pentru desfășurarea procesului de studiu

Orele la modulul *Limbaje de programare* trebuie să fie ținute în laboratorul din unitatea de învățământ, amenajat și dotat cu echipament corespunzător.

Resurse materiale minime necesare pentru parcurgerea modulului:

- calculatoare;
- conexiune la internet pentru acces la resurse educaționale și documentație online;
- sisteme de operare compatibile;

- mediu de programare (IDE) pentru limbajele studiate;
- software pentru documentare: browser web, PDF reader, Microsoft Word/LibreOffice Writer;
- manuale/suporturi de curs pentru limbajele de programare studiate;
- fișe/îndrumar de laborator și teme practice pentru toate temele disciplinei;
- resurse online: tutoriale, documentație oficială, forumuri și platforme de codare;
- proiector/ecran interactiv pentru prezentarea teoretică și demonstrații practice în laborator.

XII. Resursele didactice recomandate elevilor

Nr. crt.	Denumirea resursei	Locul unde poate fi consultată/accesată resursa
1.	Emanuela Cerchez, Marinel Serban Programarea in limbajul C/C++ pentruliceu – vol. I, II, II, IV., POLIROM	Biblioteca
2.	Nicolae Constantinescu, Programare Procedurala, Editura Universitaria, 2009	Biblioteca
3.	Doina Logofătu, Algoritmi fundamentaliin C++. Aplicații, Ed. 1, Editura Polirom, Iași, 2007	Biblioteca
4.	B. Kernigham, D. Ritchie - Limbajul deprogramare C, editia a 2-a, Ed. Teora,2003.	Biblioteca
5.	M. Cosulschi, M. Gabroveanu - Algoritmi - o abordare pragmatica, editia a 2-a, Ed. Universitaria, 2003.	Biblioteca
6.	C. Седжвик, Язык программированияСи++. М.: Мир, 2001.	Biblioteca
7.	C. Botez, M. Gospodaru, Limbaje de programare. Introducere în limbajul C.Iași, 1998.	Biblioteca
8.	N. Barkakati, Borland C++. Ghidulprogramatorului. București: Teora, 1997.	Biblioteca
9.	C. Namir, Shammass, Curs rapid deBorland C++ 4. București: Teora,1996 (trad.engl.).	Biblioteca
10.	http://www.cplusplus.com/	Internet
11.	http://infoscience.3x.ro/c++.html	Internet
12.	http://www.studytonight.com/cpp/introduction-to-cpp.php	Internet
13.	http://www.scribub.com/stiinta/informatica/c/	Internet