

MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII MOLDOVA
UNIVERSITATEA DE STAT „ALECU RUSSO” DIN BĂLȚI
FACULTATEA DE ȘTIINȚE REALE, ECONOMICE ȘI ALE MEDIULUI
CATEDRA DE MATEMATICĂ ȘI INFORMATICĂ

Curriculum pentru unitatea de curs

BAZELE PROGRAMĂRII I

Ciclul I – studii superioare de licență

Codul și denumirea domeniului general de studii: 061 Tehnologii ale informației și comunicațiilor

Specialitatea: 0613.1 Tehnologia informației

Forma de învățământ: cu frecvență

Autori:

conf. univ., dr. Nona DEINEGO

conf. univ., dr. Diana MOGLAN

BĂLȚI, 2021

Curriculum-ul a fost discutat la ședința Catedrei de Matematică și Informatică, proces-verbal nr. 8 din 12.02. 2021.

Șeful Catedrei de Matematică și Informatică, dr., conf. univ. Corina Negara

Corina Negara

Curriculum-ul a fost aprobat la ședința Consiliului Facultății de Științe Reale, Economice și ale Mediului, proces-verbal nr. 11 din 16.04. 2021.

Decanul Facultății de Științe Reale, Economice și ale Mediului, dr., conf. univ. Ina Ciobanu



Informații de identificare a unității de curs

Facultatea: Științe Reale, Economice și ale Mediului

Catedra: matematică și informatică

Domeniile generale de studiu: 061 Tehnologii ale informației și comunicațiilor

Domeniul de formare profesională la ciclul I: 0613 Dezvoltarea produselor program și a aplicațiilor

Specialitatea: 0613.1 Tehnologia informației

Denumirea unității de curs: Bazele programării I

Administrarea unității de curs:

Plan licență	Codul unității de curs	Credite ECTS	Total ore	Repartizarea orelor				Forma de evaluare	Limba de predare
				Curs	Sem.	Lab.	Lucrul ind.		
TI	F.01.O.002	6	150	44	–	46	90	Examen	Rom/Rus

Anul de studiu și semestrul în care se studiază disciplina: Anul I, Semestrul 1

Forma de organizare a învățământului: Cu frecvență

Regimul disciplinei (obligatorie / opțională / la liberă alegere): Obligatorie

Categoria formativă: Unitate de curs fundamentală.

Informații referitoare la cadrele didactice

Nona DEINEGO, doctor în științe pedagogice, conferențiar universitar. Absolventă a Universității Naționale din Sevastopol, specialitatea „Mașini Electronice de Calcul”. A susținut teza de doctor în pedagogie la specialitatea „Teoria și metodologia instruirii (Informatică)”. A realizat publicații aplicative și metodice cu tematica: „Modelul conceptual al evaluării adaptive a nivelului de pregătire al studenților”, „Modelarea matematică în testarea pedagogică”, „Testarea adaptivă computerizată bazată pe teoria IRT”. Formator permanent din anul 2003 în cadrul cursurilor de formare continuă a profesorilor de informatică.

E-mail: deinego.nona@usarb.md

Skype: [deinego.nona](https://www.skype.com/join/deinego.nona)

Orele de consultații: Marți, joi, 14:30 – 16:00. Consultațiile se oferă atât în regim „față-în-față”, cât și prin utilizarea poștei electronice sau Skype.

Diana MOGLAN, doctor în științe pedagogice, conferențiar universitar, absolventa Universității de Stat „Alecu Russo” din Bălți, specialitatea „Matematică și informatică”. A efectuat studiile de master la specialitatea „Informatică”. A susținut teza de doctor în pedagogie la specialitatea „Teoria și metodologia instruirii (științe exacte și ale mediului)”.

E-mail: mogdiana@gmail.com

Skype: mogdiana

Orele de consultații: Miercuri, 15:00 – 16:30. Consultațiile se oferă atât în regim „față-în-față”, cât și prin utilizarea poștei electronice sau prin Skype.

Integrarea unității de curs în programul de studii

Cursul „Bazele programării I” se predă în semestrul I la toate specialitățile de Informatică. Este o disciplină fundamentală. Rezolvarea unei probleme la calculator parcurge 4 etape: specificarea problemei, proiectarea rezolvării (algoritmului), codificarea rezolvării (programarea propriu-zisă), exploatarea și întreținerea problemei. Cursul are ca obiectiv familiarizarea studenților cu etapa a doua – proiectarea algoritmului.

În cadrul cursului se studiază noțiunea de date și noțiunea de algoritm, formele principale de organizare și păstrare a datelor în calculator, principalele structuri de control care se folosesc în programare: structura liniară, structura alternativă, structura repetitivă și abstracția. Cursul servește drept bază pentru disciplinele: “Bazele programării II”, “Limbaje de programare structurată”, “Programare JAVA”, “Programare Web”, “Gestiunea informației”, “Programarea orientată pe obiect” etc.

Viitorii programatori vor învăța să proiecteze diverși algoritmi, să folosească cele mai optimale metode.

Cunoștințe și competențe învățate/dobândite anterior, necesare pentru

însușirea unității de curs

1. Cunoștințe:

- cunoașterea arhitecturii și principiului de funcționare al calculatorului;
- cunoașterea reprezentării în memoria operativă a numerelor fără semn;
- cunoașterea reprezentării în memoria operativă a numerelor cu semn;
- cunoașterea reprezentării în memoria calculatorului a informației caracteriale;
- cunoașterea reprezentării numerelor negative în memoria calculatorului;
- cunoașterea particularităților tabelului ASCII.

2. Deprinderi:

- a transforma un număr întreg în binar ca număr fără semn;
- a transforma un număr întreg în binar ca număr cu semn;
- a reprezenta în binar un caracter;
- a reprezenta în binar un șir de caractere;

- a utiliza eficient resursele sistemului de calcul, sistemului de operare și ale Internetului.
3. Posedarea la nivel teoretic și aplicativ a cunoștințelor și abilităților dobândite în cadrul disciplinei preuniversitare „Informatica” în limitele standardelor de studiu eficient al învățării.

Competențele formate/dezvoltate în cadrul unității de curs

În cadrul studierii unității de curs „Bazele programării I” la studenți vor fi dezvoltate următoarele competențe (CP – competențe profesionale; CT – competențe transversale):

CP1. Îmbunătățirea proceselor. Valorifică cunoștințele de specialitate pentru a studia procesele/soluțiile existente în domeniul TIC în vederea definirii posibilelor inovații. Face recomandări bazate pe argumente științifice.

CP2. Elaborarea și proiectarea arhitecturii. Exploatează cunoștințele de specialitate pentru a defini adecvat tehnologia și specificațiile relevante necesare pentru elaborarea proiectelor TIC, a aplicațiilor sau a îmbunătățirii infrastructurii.

CP3. Proiectarea și dezvoltarea aplicațiilor. Acționează creativ pentru a dezvolta aplicații și a selecta opțiunile tehnice adecvate. Optimizează dezvoltarea, întreținerea și performanța aplicațiilor prin utilizarea modelelor de design și prin reutilizarea soluțiilor testate.

CT1. Autonomie și responsabilitate. Aplică reguli de muncă riguroasă și eficientă, manifestă atitudini responsabile față de domeniul profesional, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională.

CT2. Interacțiune socială. Identifică rolul și responsabilitățile într-o echipă plurispecializată cu aplicarea de tehnici de relaționare și muncă eficientă în cadrul echipei.

CT3. Dezvoltare personală și profesională. Identifică oportunitățile de formare continuă și valorificarea eficientă a resurselor și tehnicilor de învățare pentru propria dezvoltare.

Finalitățile unității de curs

La finalizarea studierii unității de curs și realizarea sarcinilor de învățare studentul va fi capabil:

- să utilizeze adecvat noțiunile de programare;
- să utilizeze corect metodele și modelele prezentate la curs pentru rezolvarea problemelor propuse;
- să identifice și să descrie conceptele, teoriile și metodele de bază utilizate în programare;
- să rezolve problemele practice care includ elemente de structuri de date statice și algoritmi;
- să proiecteze produse soft pentru prelucrarea datelor de tipuri elementare;

- să proiecteze produse soft pentru prelucrarea structurilor de date statice;
- să analizeze corectitudinea algoritmilor.

Conținutul unității de curs și metodele de instruire

Unitatea de curs „Bazele programării I” este divizată în două unități de învățare.

Prelegeri (P – ore prelegeri; LI – ore lucru independent)

Nr. d/o	Subiectele de studiu	Ore
		P/LI
<i>Unitatea de învățare nr. 1. Tipuri elementare de date și structuri de control – 20 ore</i>		
1.	Etapale procesului de programare. Specificarea problemei. Proiectarea algoritmului. Codificarea algoritmului. Exploatarea și întreținerea programului.	2/2
2.	Tipuri elementare de date. Date. Variabile. Tipuri elementare de date. Clasificarea tipurilor elementare după continuitatea valorilor și după necesitatea declarării.	2/2
3.	Metode de proiectare a algoritmilor. Principiile programării structurate. Esența metodei proiectării descendente a algoritmilor. Modelul descrierii algoritmului în pseudocod.	2/2
4.	Structura liniară. Modelul structurii liniare. Reprezentarea grafică a structurii liniare. Instrucțiunile utilizate în structura liniară. Verificarea corectitudinii algoritmului liniar	2/2
5.	Structura alternativă. Algoritmi alternativi. Clasificarea construcțiilor alternative. Expresii logice. Selecția duală. Selecția multiplă.	3/2
6.	Structura repetitivă. Algoritmi repetitivi. Clasificarea proceselor repetitive. Cicluri cu un număr cunoscut de repetări. Elementele ciclului. Construcții repetitive.	2/2
7.	Structura repetitivă. Cicluri cu un număr necunoscut de repetări. Elementele ciclului. Construcții repetitive.	2/2
8.	Abstracția în programare. Avantajele utilizării subprogramelor. Tipuri de subprograme. Definirea procedurii/funcției. Lista parametrilor formali. Metode de transmitere a parametrilor. Apelul procedurii/funcției. Lista parametrilor actuali. Variabile globale. Variabile locale.	4/4
9.	Abstracția datelor. Atribuirea reușită a identificatorilor. Utilizarea constantelor. Metode de formare a tipurilor de date utilizator. Metode de formare a tipurilor de date utilizator. Tipul de date enumerare. Tipul de date interval.	2/2
<i>Unitatea de învățare nr. 2. Prelucrarea structurilor de date statice – 24 ore</i>		
10.	Tablouri unidimensionale. Tipul de date tablou unidimensional. Declararea tipului. Declararea variabilelor. Accesarea elementelor. Prelucrarea secvențială a elementelor tabloului unidimensional. Metode tipice de prelucrare. Introducerea tabloului. Afișarea tabloului. Numărarea elementelor tabloului care posedă o proprietate. Însurarea elementelor tabloului. Determinarea valorii maxime (minime). Determinarea existenței în tablou a elementelor cu o proprietate. Deplasarea elementelor tabloului. Rotirea elementelor tabloului. Includerea	8/8

Nr. d/o	Subiectele de studiu	Ore
		P/LI
	elementelor în tablou. Excluderea elementelor din tablou. Sortarea tablourilor. Interclasarea tablourilor.	
11.	Prelucrarea șirurilor de caractere. Tipul de date șir de caractere. Declararea tipului. Declararea variabilelor. Accesarea elementelor. Operații cu șiruri de caractere. Prelucrarea șirurilor de caractere. Numărarea caracterelor din șir care posedă o careva proprietate. Determinarea existenței în șir a elementului cu o careva proprietate. Determinarea poziției primului caracter din șir care posedă o careva proprietate. Extragerea unui subșir dintr-un șir. Ștergerea unui subșir dintr-un șir. Includerea unui subșir într-un șir. Vectori de șiruri de caractere.	3/3
12.	Tablouri bidimensionale. Tipul de date tablou bidimensional. Declararea tipului. Declararea variabilelor. Accesarea elementelor. Prelucrarea secvențială a elementelor tabloului bidimensional. Metode tipice de prelucrare. Introducerea tabloului. Afișarea tabloului. Numărarea elementelor tabloului care posedă o proprietate. Însurarea elementelor tabloului. Determinarea valorii maxime (minime). Determinarea existenței în tablou a elementelor cu o proprietate. Includerea unui rând în tablou. Includere unei coloane în tablou. Excluderea unui rând. Excluderea unei coloane. Interschimbarea a două rânduri. Interschimbarea a două coloane.	4/4
13.	Structuri neomogene. Tipuri de articole. Articole fixe. Articole cu variante. Declararea tipului articol. Declararea variabilelor. Accesarea câmpurilor. Vectori de articole.	2/3
14.	Utilizarea mulțimilor. Proprietățile mulțimilor. Declararea mulțimilor. Domeniul de definiție a mulțimilor. Reprezentarea în memorie a mulțimilor. Operații cu mulțimile. Atribuirea. Reuniunea mulțimilor. Intersecția mulțimilor. Diferența mulțimilor. Selectarea unui element al mulțimii. Formarea mulțimilor. Afișarea mulțimilor.	2/2
15.	Utilizarea fișierelor. Modelul de organizare a calculatorului. Dispozitive principale. Dispozitive de Intrare/ Ieșire. Noțiuni de fișier. Structura fișierului. Tipuri de fișiere. Reprezentarea fișierelor în algoritm. Operații cu date de tip fișier. Asocierea fișierului cu o variabilă. Deschiderea fișierului. Scrierea componentelor în fișier. Citirea componentelor din fișier. Închiderea fișierului. Prelucrarea fișierului. Crearea fișierului. Prelucrarea fișierului.	4/4
16.	Lecție de sinteză.	1
Total		44/44

Laborator (L – ore de laborator, LI – ore lucru independent)

Nr. d/o	Subiectele de studiu	Ore
		L/LI
1.	Specificarea problemelor.	2/2
2.	Tipuri de date standard.	2/2
3.	Metode de proiectare a algoritmilor.	1/1

Nr. d/o	Subiectele de studiu	Ore
		L/LI
4.	Structura liniară.	2/2
5.	Structura alternativă.	3/3
6.	Procese repetitive cu un număr cunoscut de repetări.	4/4
7.	Procese repetitive cu condiție de terminare.	4/4
8.	Abstracția în programare.	4/4
9.	Abstracția datelor.	1/1
10.	Prelucrarea tablourilor unidimensionale.	6/6
11.	Prelucrarea șirurilor de caractere.	2/2
12.	Prelucrarea tablourilor bidimensionale.	4/5
13.	Prelucrarea structurilor neomogene.	4/4
14.	Utilizarea mulțimilor.	2/2
15.	Prelucrarea fișierelor.	4/4
16.	Lecție de sinteză.	1
Total		46/46

Pentru unitatea de curs este creat un suport de susținere a activităților la curs pe platforma MOODLE, disponibil pe <http://old.elearning.usarb.md/moodle/course/view.php?id=164>. Odată cu începerea cursului studenții sunt înregistrați la curs pe platforma MOODLE și trebuie să realizeze diferite sarcini propuse pe platformă (studierea materialului, realizarea testelor grilă, rezolvarea problemelor).

La predarea conținutului unității de curs respective vor fi aplicate următoarele metode de predare și învățare: expunerea, conversația, algoritmizarea, exercițiul și problematizarea.

Pentru asigurarea realizării metodelor didactice menționate se vor utiliza suportul de curs, cursul de instruire pe platforma universitară, culegere de prezentări Power Point, consultații individuale.

Activități de lucru individual

Activitatea de lucru individual este o componentă obligatorie a activității de învățare la disciplină. În cadrul studierii acestui curs studenților li se vor propune o serie de teme și probleme care urmează a fi studiate și soluționate independent. Însărcinările practice pentru lucrul individual sunt propuse în cadrul orelor de laborator.

Studentul trebuie să realizeze un set de probleme cu însărcinări practice la 14 lucrări de laborator, unde sunt indicate: scopul lucrării, formularea problemei, date inițiale, indicații metodice

și recomandări, cerințele față de program și variantele însărcinărilor. În special sunt prezente sarcini de implementare a materialului teoretic cu aplicații în practică.

Activitatea individuală pe parcursul semestrului va fi apreciată cu notă la finele semestrului.

Exemple de însărcinări practice pentru lucrul individual:

1. *Specificarea problemei.* Elaborați specificarea următoarei probleme: "Calcularea notei medii a clasei, dacă la o lucrare de control elevii au primit Z note de 10, N de 9, O de 8, S de 7 și P de 4".
2. *Structura liniară.* Se citește un număr natural X format din 4 cifre. Să se elaboreze un program care afișează numărul obținut din suma numerelor care se obțin eliminând pe rând prima cifră a lui X, apoi primele două, apoi primele trei. Exemplu: Dacă $X=2347$ se obține $347+47+7=40$.
3. *Structura alternativă. Construcția If.* Ionel are înălțimea H1 cm, Gigel are H2 cm, iar Dănuț are H3 cm. Să se elaboreze un program care afișează numele celor 3 copii în ordinea crescătoare a înălțimii.
4. *Structura alternativă. Construcția Case.* Unitățile de măsură se numerotează în felul următor (1 - kg, 2 - mg, 3 - g, 4 - tona). Se introduce de la tastieră greutatea unui corp și numărul unității de măsură. Să se elaboreze un program care afișează greutatea corpului în kg.
5. *Structura repetitivă. Ciclul cu contor.* Se dă un număr natural n. Să se elaboreze un program care îl descompune și afișează, dacă este posibil, ca sumă a două numere consecutive. Exemplu: Date de intrare: n=5; Date de ieșire: $5=2+3$; Date de intrare: n=6; Date de ieșire: Nu e posibil.
6. *Structura repetitivă. Ciclul cu condiție de terminare.* Se introduc de la tastatură un șir de numere întregi până la 0 (nenule). Să se elaboreze un program care determină numărul elementelor de trei cifre.
7. *Subprograme. Proceduri.* Să se elaboreze o procedură care determină dacă un număr se divide la toate cifrele sale. Utilizând procedura să se afișeze toate numerele din intervalul 20..30 care verifică această proprietate.
8. *Subprograme. Funcții.* Să se elaboreze un program care citește de la tastatură un număr natural nenul din cel mult 4 cifre n și prin apeluri ale funcției PERFECT verifică dacă n este pătrat perfect. Programul va afișa pe ecran în caz afirmativ mesajul DA, iar în caz contrar mesajul NU.
9. *Tablouri unidimensionale.* Să se elaboreze un program care completează un vector A cu n numere întregi citite de la tastieră și ordonează descrescător elementele vectorului până la poziția k (citită de la tastieră) și crescător restul vectorului. La final programul afișează pe ecran vectorul obținut.
10. *Tablouri unidimensionale.* Pentru n elevi ($n \leq 25$) dintr-o clasă se cunosc mediile anuale la informatică. Să se elaboreze un program care completează un vector A cu mediile anuale la

informatică ale elevilor și determină dacă mediile anuale la informatică ale elevilor sunt aranjate în ordinea descrescătoare a acestor medii.

11. *Șiruri de caractere*. Se citește un text de cel mult 255 de caractere, format din mai multe cuvinte separate prin spații care se termină cu caracterul ‘.’. Să se elaboreze un program care afișează cuvântul din text cu lungime maximă.
12. *Tablouri bidimensionale*. Se citește o matrice A cu n linii și m coloane ($m \leq 30$, $n \leq 30$) cu elemente numere întregi. Să se elaboreze un program care determină indicii elementelor pentru care suma vecinilor (stânga, dreapta, sus, jos) are valoare pozitivă.
13. *Structuri neomogene. Articole*. Un magazin alimentar conține produse ce definesc o structură cu următoarele câmpuri: denumire, preț și cod numeric. Să se elaboreze un program care formează un tabel cu informații despre n ($n \leq 100$) produse și:
 - să se majoreze prețurile cu 5% în cazul în care codul produselor este mai mic sau egal cu 200 și cu 13% dacă este mai mare decât 200.
 - să se afișeze noile prețuri și denumirile produselor;
 - să se afișeze produse cu prețul cel mai mare.
14. *Fișiere*. Se consideră structura:

Număr inventar	Identificare carte				Nr. bucăți în inventar
	Autori	Titlu	Editură	An apariție	

- 14.1. Să se construiască un fișier cu structura de mai sus, cu preluarea informațiilor de la tastatură.
- 14.2. Să se listeze conținutul acestui fișier.
- 14.3. Să se determine cel mai frecvent an de apariție a volumelor din bibliotecă.

Evaluarea

Cunoștințele, abilitățile și competențele studenților vor fi evaluate pe parcursul semestrului cu trei note:

- a) nota evaluării curente N_{ec} care va fi determinată prin calcularea mediei notelor obținute în cadrul orelor de laborator prin rezolvări de probleme, argumentări a elaborării algoritmului de calcul (fiecare student trebuie să fie evaluat cu cel puțin 4 note) și în cadrul orelor de prelegeri prin realizarea a 2 sarcini practice, promovate sub formă de test grilă cu itemi teoretici și practici pe platforma MOODLE;
- b) nota evaluării periodice N_p care va fi determinată după studiul a jumătate din partea teoretică a cursului prin susținerea unui test grilă cu itemi teoretici și practici pe platforma MOODLE. Nota obținută se rotunjește conform legilor matematice;

c) nota evaluării lucrului individual N_i .

La evaluarea finală sunt admiși studenții care întrunesc următoarea condiție – toate notele N_{ec}, N_p, N_i au valoarea de cel puțin 5.

Nota medie semestrială N_s se calculează ca media aritmetică a componentelor de mai sus conform formulei:

$$N_s = (N_{ec} + N_p + N_i)/3.$$

Nota medie semestrială constituie 60% din nota generală la unitatea de curs. Celelalte 40% le constituie nota de la examen.

Evaluarea finală este realizată prin susținerea unui test grilă cu 25 itemi teoretici și practici pe platforma MOODLE. Durata administrării testului este de 90 min.

Nota generală N_g la unitatea de curs „Bazele programării I” se calculează (cu două cifre zecimale după virgulă) conform formulei:

$$N_g = 0,6 \times N_s + 0,4 \times N_e,$$

unde N_e este nota de la examen.

Modele de teste grilă de evaluare curentă, periodică și finală sunt plasate pe platforma de învățare MOODLE.

VARIANTA I
Bazele programării I
Testul conține 14 itemi

№	ÎNSĂRCINAREA	Puncte								
1.	<p>Se consideră următoarele declarații:</p> <p><i>Var</i></p> <p><i>S: Char</i></p> <p><i>N, K: Real</i></p> <p><i>M: Integer</i></p> <p><i>Procedure Test (Var A:Char, B:Real, Var C:Real, Var Z:Integer)</i></p> <p>Indică, instrucțiunea de apel a procedurii Test scrisă corect.</p> <table border="1" data-bbox="272 898 1198 1025"> <thead> <tr> <th data-bbox="272 898 488 943">1</th> <th data-bbox="488 898 715 943">2</th> <th data-bbox="715 898 903 943">3</th> <th data-bbox="903 898 1198 943">4</th> </tr> </thead> <tbody> <tr> <td data-bbox="272 943 488 1025"><i>Test(S,N,5,K,M)</i></td> <td data-bbox="488 943 715 1025"><i>Test('E',10,K,M)</i></td> <td data-bbox="715 943 903 1025"><i>Test(S,19,K,M)</i></td> <td data-bbox="903 943 1198 1025"><i>Nici un răspuns nu este corect</i></td> </tr> </tbody> </table>	1	2	3	4	<i>Test(S,N,5,K,M)</i>	<i>Test('E',10,K,M)</i>	<i>Test(S,19,K,M)</i>	<i>Nici un răspuns nu este corect</i>	3
1	2	3	4							
<i>Test(S,N,5,K,M)</i>	<i>Test('E',10,K,M)</i>	<i>Test(S,19,K,M)</i>	<i>Nici un răspuns nu este corect</i>							
2.	<p>Fie dat următorul fragment:</p> <p><i>Var V1, V2, V3, V4 : Integer</i></p> <p><i>Procedure P1 (F1, F2: Integer, Var F3, F4: Integer)</i></p> <p><i>Var X: Integer</i></p> <p><i>Begin</i></p> <p><i>F2 := F3</i></p> <p><i>F3 := F4 - F2</i></p> <p><i>X := F1 * F2</i></p> <p><i>F3 := F3 div 2</i></p> <p><i>F1 := X + F2</i></p> <p><i>WriteNat(F1, F2, F3, F4)</i></p> <p><i>Writeln</i></p> <p><i>End</i></p> <p><i>Begin</i></p> <p><i>V1 := 2</i></p> <p><i>V2 := 3</i></p> <p><i>V3 := 1</i></p> <p><i>V4 := 4</i></p> <p><i>P1(V1, V2, V3, V4)</i></p> <p><i>WriteNat(V1, V2, V3, V4)</i></p> <p><i>End</i></p> <p>Indică, ce se va afișa pe ecran în rezultatul executării acestui algoritm.</p>	4								
3.	<p>Fie dat următorul fragment:</p> <p><i>S:=0</i></p>	3								

№	ÎNSĂRCINAREA	Puncte
	<p>For i:=1 to 2 step 1 For j:=4 to 6 step 1 For k:=6 to 5 step -1 S:=S+1 End End End</p> <p>Indică care va fi valoarea variabilei S după executarea acestui fragment.</p>	
4.	<p>Se consideră algoritmul:</p> <p>Var A, B: Integer</p> <p>Procedure Calcul (A: Integer, Var B: Integer)</p> <p>Begin A := 5 B := -4 End</p> <p>Begin A := 7 B := 3 Calcul(A,B) WriteInt(A*B) End</p> <p>Indică, ce se va afișa pe ecran în rezultatul executării acestui algoritm.</p>	3
5.	<p>Se consideră următorul fragment:</p> <p>Var Y: Integer</p> <p>Function F1 (Var A: Integer): Integer</p> <p>Begin A := A - 1 Return A End</p> <p>Begin Y:= 3 WriteInt(Y+ F1(Y)+Y) End</p> <p>Indică ce va afișa acest algoritm.</p>	3
6.	<p>Se consideră următorul fragment:</p> <p>Type Vector = array[4..12] of Integer</p> <p>Var S: Vector</p> <p>Function F2(Var V: Vector):Integer</p> <p>Var i : Natural R : Integer</p>	3

№	ÎNSĂRCINAREA	Puncte
	<pre> Begin R:=0 i:=5 While i<12 do R := R + V[i] I := Succ(i) End Return R End </pre> <p>Indică ce va returna apelul F2(S), dacă vectorul S are următorul conținut: S=(9,10,-1,14,9,-3,7,16,-5)</p>	
7.	<p>Se consideră următorul fragment:</p> <pre> B := False R := 45 While Not B do B := R=15 R := (R mod 4) +15 End </pre> <p>Indică de câte ori se execută corpul acestui ciclu.</p>	3
8	<p>Se consideră următorul fragment:</p> <pre> X := 3 Y := 2 Repeat Y := Y + X X := Succ(X) Until X>4 X := X + Y </pre> <p>Indică care va fi valoarea variabilei X după executarea acestui fragment.</p>	4
9	<p>Se consideră următorul fragment:</p> <pre> Type Semnal = (Rosu, Galben, Verde) Var A: Semnal Begin A:=Verde A:=Pred(A) Case A of Rosu: WriteString ('Stop') Galben: WriteString ('Atentie') Verde: WriteString ('Start') End End </pre> <p>Indică, ce va afișa acest algoritm.</p>	2
10	<p>Se consideră următorul algoritm:</p> <pre> Type </pre>	4

№	ÎNSĂRCINAREA	Puncte												
	<p><i>Tb = array[1..3, 1..4] of Integer</i></p> <p><i>Var X: Tb</i></p> <p><i>Function F3 (A : Tb):Integer</i></p> <p><i>Var i, j, R : Integer</i></p> <p><i>Begin</i></p> <p><i> R := A[1,2]</i></p> <p><i> j := 2</i></p> <p><i> For i:=2 to 3 step 1</i></p> <p><i> If A[i,j] < R then</i></p> <p><i> R := A[i,j]</i></p> <p><i> End</i></p> <p><i> End</i></p> <p><i> Return R</i></p> <p><i>End</i></p> <p>Fie că variabila X are următorul conținut:</p> <table border="1" data-bbox="272 853 576 972"> <tr> <td>2</td> <td>6</td> <td>1</td> <td>8</td> </tr> <tr> <td>4</td> <td>6</td> <td>7</td> <td>3</td> </tr> <tr> <td>4</td> <td>6</td> <td>6</td> <td>6</td> </tr> </table> <p>Indică, ce valoare va returna apelul funcției F3(X).</p>	2	6	1	8	4	6	7	3	4	6	6	6	
2	6	1	8											
4	6	7	3											
4	6	6	6											
11	<p>Considerăm următorul algoritm:</p> <p><i>Type</i></p> <p><i> Sir = String[50]</i></p> <p><i>Var X: Sir</i></p> <p><i>Procedure P2 (S : Sir)</i></p> <p><i> Var i : Natural</i></p> <p><i> Begin</i></p> <p><i> i := 2</i></p> <p><i> While (i <= length(S)) Do</i></p> <p><i> WriteChar(S[i])</i></p> <p><i> i := i + 2</i></p> <p><i> End</i></p> <p><i> End</i></p> <p><i> Begin</i></p> <p><i> X := 'Abstract'</i></p> <p><i> P2(X)</i></p> <p><i> End</i></p> <p>Indică, ce se va afișa pe ecran în rezultatul executării acestui algoritm.</p>	3												
12	<p>Se consideră următorul algoritm:</p> <p><i>Type</i></p> <p><i> T1=Array[1..9] of Integer</i></p> <p><i>Var</i></p> <p><i> S:T1</i></p> <p><i>Function F1(V:T1):Integer</i></p>	3												

№	ÎNSĂRCINAREA	Puncte
	<pre> Var I: Natural M: Integer Begin M:=V[1] I:=2 While I<9 do If V[I]<=M then M:= V[I] End I:=succ(I) End Return M End Begin {formarea vectorului} WriteNat(F1(S)) End Indică ce va afișa pe ecran apelul F1(S), dacă vectorul S are următorul conținut S=(5,7,5,9,1,12,1) </pre>	
13	Scrieți o funcție logică care determină dacă un caracter este vocală.	5
14	Să se elaboreze un program care afișează numărul de cifre a unui număr natural.	5

Barem de notare

Nota	1	2	3	4	5	6	7	8	9	10
Punctele	1-4	5-9	10-15	16-20	21-24	25-30	31-35	36-41	42-45	46-48

Resurse informaționale la unitatea de curs

Obligatorii:

1. CABAC, V., DEINEGO, N. *Bazele programării I. Curs fundamental. Vol. 1: Tipuri elementare de date și structuri de control*. Bălți: Presa universitară Bălțeană, 2013. 222 p.
2. CABAC, V., DEINEGO, N., SKUTNIŢKI, O. *Bazele programării II. Curs fundamental. Vol. 2: Prelucrarea structurilor statice de date*. Bălți: Tipografia din Bălți, 2019. 213 p.
3. DEINEGO, N., CABAC, V., MOGLAN, D. *Основы программирования: Базовый курс. Vol. I: Элементарные типы данных и управляющие структуры*. Bălți: Presa universitară Bălțeană, 2016, 234 p.
4. DEINEGO, N., CABAC, V., MOGLAN, D. *Основы программирования: Фундаментальный курс. Vol. II: Обработка статических структур данных*. Bălți: Tipografia din Bălți, 2019, 222 p.
5. РАЙЛИИ, Д. *Использование языка Модула-2. Вводный курс*. Москва: изд-во Мир, 1993. 606 с.
6. ВИРТ, Н. *Алгоритмы и структуры данных*. СПб.: изд-во Невский диалект, 2001. 352 с.

Suplimentare:

1. THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD R. RIVEST. *Introducere în algoritmi*. Cluj: ed. Libris Agora, 2000. 880 p.
2. PĂTRĂȘCOIU, O., MARIAN, Gh., MITROI, N. *Elemente de grafuri și combinatorică. Metode, algoritmi și programe*. București: ed. All, 1994. 224 p.

Principiile de lucru în cadrul unității de curs

1. Calendarul cursului (termenii-limită de prezentare a sarcinilor propuse spre rezolvare, momentele de evaluare etc.) este corelat cu calendarele la alte discipline din semestru. De aceea prezentarea sarcinilor după termenul-limită indicat în calendar nu este salutăată, iar studenții care amână frecvent prezentarea sarcinilor își formează o imagine nefavorabilă.
2. Nu este salutăată întârzierea la ore.
3. Este salutăată poziția activă a studentului, care studiază din propria inițiativă noi conținuturi, propune soluții, formulează întrebări în cadrul prelegerilor și a orelor practice.
4. În cadrul disciplinei o atenție sporită va fi oferită respectării principiilor etice. Prezentarea unor soluții a sarcinilor, preluate de la colegi sau din alte surse, preluarea informațiilor din diverse surse, fără a face trimitere la sursă, va fi considerată plagiat și va fi sancționată prin note de „1”.
5. Înregistrarea pe platforma de învățare MOODLE și înscrierea la cursul electronic ”Bazele programării I” este obligatorie.