

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 II

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



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



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 II” se predă în semestrul II 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 structurile de date dinamice și metodele de proiectare a algoritmilor. Cursul servește drept bază pentru disciplinele: “Limbaje de programare structurată”, “Programare JAVA”, “Programare Web”, “Gestiunea informației”, “Programarea orientată pe obiect” etc.

Viitorii programatori și profesori de informatică 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 conceptelor unității de curs ”Bazele programării I”.
2. Deprinderi:
 - proiectarea algoritmilor pentru prelucrarea datelor elementare;
 - proiectarea algoritmilor pentru prelucrarea structurilor de date statice;
 - utilizarea eficientă a resurselor sistemelor de calcul, 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 II” 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 modelele și metodele de programare prezentate la curs pentru rezolvarea problemelor propuse;
- să rezolve problemele practice care includ elemente de structuri de date dinamice și algoritmi;
- să gestioneze eficient memoria internă a calculatorului;
- să proiecteze produse soft pentru prelucrarea structurilor de date dinamice;
- să proiecteze produse soft pentru prelucrarea structurilor de date arborescente;
- să analizeze corectitudinea algoritmilor.

Conținutul unității de curs și metodele de instruire

Unitatea de curs „Bazele programării II” este divizată în trei 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. Recursivitate și metode de proiectare a algoritmilor – 12 ore</i>		
1.	Recursivitate. Noțiuni recursive. Realizarea recursivității în programare. Mecanismul recursivității. Subprograme recursive. Implementarea recursivă a algoritmilor elementari. Avantajele și dezavantajele recursivității.	2/2
2.	Metode nerecursive de proiectare a algoritmilor. Metoda Greedy. Probleme pentru care metoda Greedy conduce la soluția optimă. Suma maximă. Problema planificării spectacolelor. Problema rucsacului (cazul continuu). Problema comis-voiajorului. Programare dinamică. Problema triunghiului. Subșir crescător de lungime maximă. O problemă cu sume. Problema rucsacului (cazul discret).	4/4
3.	Metode recursive de proiectare a algoritmilor. Metoda Backtracking. Implementarea metodei Backtracking. Probleme rezolvabile prin metoda Backtracking. Generarea permutărilor. Generarea aranjamentelor. Generarea combinațiilor. Generarea tuturor partițiilor unui număr natural. Problema celor n dame. Metoda „Divide et Impera“. Implementarea metodei „Divide et Impera“. Căutarea binară. Sortarea prin interclasare. Sortarea rapidă. Problema turnurilor din Hanoi.	6/6
<i>Unitatea de învățare nr. 2. Date dinamice. Prelucrarea listelor liniare – 17 ore</i>		
4.	Gestionarea datelor dinamice. Tipul de date referință și variabile dinamice. Operații cu date de tip referință. Cazurile de apariție a nodurilor inaccesibile și a referințelor periculoase.	3/3
5.	Structuri dinamice liniare. Stiva. Prelucrarea stivei. Operații cu stiva. Crearea stivei vide. Adăugarea unui element în vârful stivei. Ștergerea unui element din vârful stivei. Parcurgerea elementelor stivei.	2/2
6.	Structuri dinamice liniare. Coadă. Prelucrarea cozii. Operații cu coada. Crearea cozii vide. Adăugarea unui element în sfârșitul cozii. Ștergerea unui element din vârful cozii. Parcurgerea elementelor cozii. Coadă cu santinelă. Coadă cu două santinele. Operații cu coada cu santinele.	4/4
7.	Structuri dinamice liniare. Lista liniară oarecare. Operații cu lista liniară oarecare. Crearea listei vide. Adăugarea unui element în listă. Ștergerea unui element din listă. Parcurgerea elementelor listei. Lista circulară.	4/4
8.	Structuri dinamice liniare. Lista bidirecțională. Operații cu lista bidirecțională. Crearea listei vide. Adăugarea unui element în listă. Ștergerea unui element din listă. Parcurgerea elementelor listei.	2/2

Nr. d/o	Subiectele de studiu	Ore
		P/LI
9.	Structuri dinamice liniare. Aplicarea listelor liniare.	2/2
<i>Unitatea de învățare nr. 3. Date dinamice. Prelucrarea structurilor arborescente de date – 15 ore</i>		
10.	Structuri ierarhice. Arbori binari. Implementarea dinamică a arborilor binari. Operații cu arborii binari. Crearea arborelui vid. Crearea rădăcinii. Crearea fiului stâng a unui nod. Crearea fiului drept a unui nod.	2/2
11.	Parcurgerea arborelui binar. Metode de parcurgere a arborilor binari. Parcurgerea arborelui binar în preordine, inordine și postordine.	2/2
12.	Construirea arborilor binari. Crearea arborilor introduși de la tastatură. Crearea unui arbore perfect echilibrat.	2/2
13.	Arbori binari de căutare. Construirea unui arbore binar de căutare. Prelucrarea arborilor binari de căutare. Determinarea valorii minime în arborele binar de căutare. Ștergerea unui nod în arborele binar de căutare.	4/4
14.	Arbori binari ai expresiilor. Proprietățile unui arbore binar a expresiei. Algoritm de calculare a valorii expresiei aritmetice reprezentată în forma poloneză inversă.	2/2
15.	Aplicarea arborilor binari. Arbori binari de sortare.	2/3
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.	Mecanismul recursivității.	4/4
2.	Metode nerecursive de proiectare a algoritmilor.	4/4
3.	Metode recursive de proiectare a algoritmilor.	4/4
4.	Gestionarea datelor dinamice.	4/4
5.	Prelucrarea stivei și a cozii.	4/4
6.	Prelucrarea listei cu două santinele.	4/4
7.	Prelucrarea listelor bidirecționale.	2/2
8.	Aplicarea listelor liniare.	3/3
9.	Implementarea dinamică a arborilor binari.	4/4
10.	Metode de parcurgere a arborilor binari.	2/2
11.	Construirea arborilor binari. Operații cu arborii binari.	4/4

Nr. d/o	Subiectele de studiu	Ore
		L/LI
12.	Prelucrarea arborilor binari de căutare.	2/2
13.	Arbori binari ai expresiilor.	2/2
14.	Aplicarea arborilor binari.	2/3
15.	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=167>. 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 6 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. *Recursivitate*. Elaborați algoritmul recursiv și algoritmul iterativ pentru rezolvarea următoarei probleme: De la tastatură se introduc un șir de caractere S și un caracter C. Să se elaboreze un program care determină de câte ori apare caracterul C în șirul de caractere S.
2. *Gestionarea datelor dinamice*. Fie date următoarele declarații:

Type

Ref = Pointer to Char

Var

PA, PB, PC, PD: Ref

Considerând ca figura de mai jos reprezintă structura memoriei înainte de executarea fiecărui fragment de mai jos, desenează structura memoriei după executarea fragmentului.



- a. $PC := PA$
 - b. $PD := PA$
 - c. $Allocate(PC, Size(Char))$
 $PC^{\wedge} := „X”$
 $PB := PC$
 - d. $PC := PA$
 $PB^{\wedge} := 'C'$
 $Deallocate(PA, Size(Char))$
 $PA := Nil$
 - e. $PC := PD$
 $PD := PB$
 $PB := PA$
 $PA := Nil$
 - f. $PC := PB$
 $PA := PC$
 $PD := PA$
 $Deallocate(PA, Size(Char))$
 - g. $Allocate(PA, Size(Char))$
 $PB^{\wedge} := 'W'$
 $PD^{\wedge} := 'Q'$
3. *Structuri dinamice de date. Liste simplu înlănțuite.* Creați listă liniară simplu înlănțuită Cărți ce conține înregistrări cu următoarea informație: numărul cărții, denumirea cărții, numele autorului, anul ediției, numărul de exemplare. Să se elaboreze un program care determină, dacă în listă există cartea, denumirea căreia se introduce de la tastatură.
4. *Structuri dinamice de date. Liste liniare speciale.* Pe o masă se găsesc n stive de farfurii colorate diferit. Se cere să se realizeze o singură stivă din cele n în așa fel încât să alterneze culorile farfuriilor (în număr egal). Să se elaboreze un program care tipărește culorile stivei rezultat. Exemplu: trei stive de culorile roșu, galben și albastru cu câte 6, 3 și respectiv 5 farfurii, va

rezultă o stivă de 9 farfurii astfel: roșu, galben, albastru, roșu, galben, albastru, roșu, galben și albastru.

5. *Structuri dinamice de date. Arbori binari.* Pentru fiecare articol dintr-o magazie, se păstrează următoarele informații (utilizând arborele binar de căutare): codul articolului (este unic pentru fiecare articol), denumirea articolului, unitatea de măsură și cantitatea din stoc. Să se elaboreze un program care să realizeze următoarele:
- Adăugarea unui nou articol;
 - Ștergerea unui articol;
 - Introducerea de la tastatură a cantității intrate, dintr-un articol precizat prin cod, și actualizarea stocului;
 - Afișarea articolelor care au stocul mai mare de 100;
 - Afișarea în ordinea crescătoare a codurilor articolelor.
6. *Metode de proiectare a algoritmilor. (Backtracking)* La un concurs sportiv s-au înscris n concurenți având numerele de concurs 1, 2, ..., n . Pentru fiecare sportiv se cunoaște țară de origine (șir de caractere). În prima zi vor intra în concurs m concurenți. Să se scrie un program care să afișeze toate posibilitățile de a stabili ordinea intrării în concurs a celor m concurenți respectând următoarele condiții:
- 2 sportivi din aceeași țară nu pot evolua unul după altul;
 - trebuie respectată ordinea crescătoare a numerelor de concurs ale sportivilor.

Evaluarea

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

- 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;
- 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, pusă de platformă, se rotunjește conform legilor matematice;
- 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 20 itemi teoretici și practici pe platforma MOODLE. Durata administrării testului este de 80 min.

Nota generală N_g la unitatea de curs „Bazele programării II” 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.

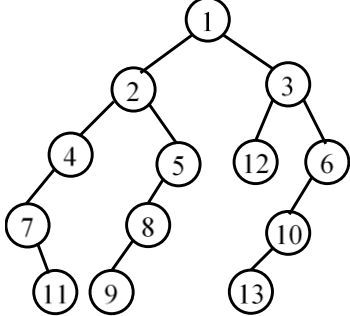
Modele de teste grilă de evaluare curentă, periodică și finală sunt plasate pe platforma de învățare MOODLE.

VARIANTA I
Bazele programării II
Testul conține 14 itemi

№	ÎNSĂRCINAREA	Puncte
	<p>Pentru însărcinările următoare, dacă nu se specifică semnificația variabilelor, se folosesc următoarele date:</p> <p>Type</p> <p><i>Lista = Pointer to Nod</i></p> <p><i>Nod = Record</i></p> <p><i>inf: Integer</i></p> <p><i>urm: Lista</i></p> <p><i>end</i></p> <p>Var <i>Prim, Prim1, Prim2, P, R, Q, St1, St2: Lista</i></p>	
1.	<p>Fie dat următorul algoritm:</p> <p>Var</p> <p><i>A: Pointer to Char</i></p> <p><i>B, C : Pointer to Integer</i></p> <p>Begin</p> <p><i>Allocate(A, size(Char))</i></p> <p><i>A^ := 'A'</i></p> <p><i>Allocate(C, size(Integer))</i></p> <p><i>C^ := 18</i></p> <p><i>B := C</i></p> <p><i>B^ := B^ + 2</i></p> <p><i>Allocate(C, size(Integer))</i></p> <p><i>C^ := -10</i></p> <p><i>A^:= Succ(A^)</i></p> <p><i>WriteInt(B^, A^, C^)</i></p> <p>End</p> <p>Desenează structura dinamică și indică ce se va afișa pe ecran în rezultatul executării acestui algoritm.</p>	3
2.	<p>În secvență următoare Prim este un pointer către primul nod al unei liste simplu înlănțuite de numere întregi. Indică, ce se va afișa pe ecran în rezultatul executării acestui algoritm, dacă numerele din lista sunt (-5, 8, 6, 8, 8, -8). Explică răspunsul.</p> <p>Procedure P1 (Prim: Lista)</p> <p>Var <i>a, z: Integer</i></p> <p>Begin</p> <p><i>a:=0</i></p> <p><i>z:=8</i></p>	3

№	ÎNSĂRCINAREA	Puncte
	<pre> While Prim<>Nil do If Prim^.inf = z then a := a+1 end Prim := Prim^.urm end WriteInt(a) End </pre>	
3.	<p>Ce structură dinamică de date reprezintă <i>coada</i>? Scrie fragmentul algoritmului pentru adăugarea unui element în coadă.</p>	4
4.	<p>Fie <i>Prim1</i> și <i>Prim2</i> doi pointeri către primele noduri ale următoarelor liste de tip coadă: (1, 3, 7, 2) și (7, 3, 1, 2, 1, 3, 2, 3). Indică, ce se va afișa pe ecran în rezultatul executării acestui algoritm. Explică răspunsul.</p> <pre> Procedure P2 (Prim: Lista) Var P, Q: Lista a: Integer Begin P := Prim1 While (P<>Nil) Do a := 0 Q := Prim2 While (Q<>Nil) Do If (Q^.inf = P^.inf) then a:=a+1 end Q := Q^.urm end WriteInt(a) P:=P^.urm end End </pre>	4
5.	<p>Elaborează o procedură care creează o listă liniară (coada) din 5 elemente în care, fiecare nod, pe lângă informația de adresă, va conține datele de tip <i>Elev</i>. Procedura va returna adresa primului nod al listei. Datele se citesc de la tastatură: <i>numele</i>: 20 de caractere; <i>prenumele</i>: 20 de caractere; un vector de numere reale cu 3 componente care rețin <i>notele elevului</i>.</p>	6
6.	<p>Dacă <i>Prim</i> este un pointer către primul nod al unei liste liniare simplu înlănțuite cu santinelă, iar numerele din noduri sunt (10, 24, 35, 61, 28, 12, 77), în această ordine de la stânga la dreapta, indică, ce se va afișa pe ecran în rezultatul executării acestui algoritm?</p> <pre> Procedure P3 (Prim: Lista) Var P, Q: Lista Begin Q := Prim </pre>	4

№	ÎNSĂRCINAREA	Puncte
	<pre> P := Prim^.urm While P <> Nil do If P^.inf mod 2 = 0 then Q^.urm := P^.urm Deallocate(P, size(Nod)) P := Q^.urm Else Q := P P := P^.urm end end Q := Prim^.urm While Q <> Nil do WriteInt(Q^.inf) Q := Q^.urm end End </pre> <p>Reprezintă grafic operațiile efectuate.</p>	
9.	<p>Fie dată următoarea funcție recursivă:</p> <pre> Function F1 (N: Natural):Natural Begin If N = 3 then Return 1 Else Return (3*N-1)+F1(N-2) End End </pre> <p>Indică ce valoare va returna apelul F1(5). Elaborează schema apelurilor recursive.</p>	4
10.	<p>Fie dată următoarea procedură recursivă:</p> <pre> Procedure P4 (N: Integer) Begin WriteInt(N) Writeln If N<50 then P4(2*N) End WriteInt(N) Writeln End </pre> <p>Indică ce va afișa pe ecran apelul P4(5). Elaborează schema apelurilor recursive.</p>	4
11.	<p>Fie dată următoarea procedură recursivă:</p> <pre> Procedure P5 (N: Integer) </pre>	4

№	ÎNSĂRCINAREA	Puncte
	<pre> Begin If (N<>1) Then WriteInt (N) If N mod 2=0 Then P5(N div 2) Else P5(3*N+1) End Else WriteInt(1) End End </pre> <p>Indică ce va afișa pe ecran apelul P5(5). Elaborează schema apelurilor recursive.</p>	
12	<ol style="list-style-type: none"> 1. Indică valoarea nodului rădăcină; 2. Enumeră valorile nodurilor terminale; 3. Scrie lista nodurilor, obținută în rezultatul parcurgerii arborelui în preordine; 4. Scrie lista nodurilor, obținută în rezultatul parcurgerii arborelui în inordine; 5. Scrie lista nodurilor, obținută în rezultatul parcurgerii arborelui în postordine. 	4
13	<p>Desenează arborele binar de căutare, care se formează în rezultatul introducerii valorilor nodurilor care se citesc de la tastatură în următoarea ordine: 25, 27, 15, -5, 10, 20, 30, 18, -1, 7, 40, 100, 22</p>	4
14	<p>Se consideră că trebuie transportate cu ajutorul unui rucsac de capacitate 34 kg, obiecte cu greutatea 20 kg, 14 kg, 12 kg, 8 kg și 6 kg. Pentru fiecare obiect transportat câștigul obținut este 15 lei, 10 lei, 25 lei, 20 lei și 12 lei. Știind că obiectele se pot încărca parțial în sac, atunci soluția optimă este? Explică răspunsul.</p> <p>Eficiență (obiecte): _____</p> <p>Câștigul: _____</p>	4

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. РАЙЛИ, Д. *Абстракция и структуры данных. Вводный курс*. Москва: изд-во Мир, 1993. 606 с.
6. ВИРТ, Н. *Алгоритмы и структуры данных*. СПб.: изд-во Невский диалект, 2001. 352 с.
7. TUDOR, S. *Informatică: manual pentru clasa a X-a*. București: Editura L&S Infomat, 2009, 270 p.

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 salutată, iar studenții care amână frecvent prezentarea sarcinilor își formează o imagine nefavorabilă.
2. Nu este salutată întârzierea la ore.
3. Este salutată 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 II" este obligatorie.