

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babes-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica si Informatica
1.3 Departamentul	Departamentul de informatica
1.4 Domeniul de studii	Matematica
1.5 Ciclul de studii	Licenta
1.6 Programul de studiu / Calificarea	Matematica Informatica – limba de studiu romana

2. Date despre disciplină

2.1 Denumirea disciplinei	Fundamentele Programării						
2.2 Titularul activităților de curs	Conf. Dr. Diosan Laura						
2.3 Titularul activităților de seminar	Conf. Dr. Diosan Laura						
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6. Tipul de evaluare	C	2.7 Regimul disciplinei	Obligatorie

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	6	Din care: 3.2 curs	2	3.3 seminar/laborator	2/2
3.4 Total ore din planul de învățământ	84	Din care: 3.5 curs	28	3.6 seminar/laborator	28/28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					7
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					21
Tutoriat					4
Examinări					20
Alte activități:					-
3.7 Total ore studiu individual		66			
3.8 Total ore pe semestru		150			
3.9 Numărul de credite		6			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	- Noțiuni elementare de matematică din liceu
4.2 de competențe	- Capacitate de abstractizare, gândire logică și algoritmică

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	videoproiector
5.2 De desfășurare a seminarului/laboratorului	lab.: 12 calculatoare (stații de lucru)

6. Competențele specifice acumulate

Competențe profesionale	- Abilitatea de a înțelege, de a modela și rezolva probleme reale. - Abilitatea de a analiza, de a sintetiza și de a comunica oral și în scris. - Înțelegerea și utilizarea conceptelor de bază din algoritmică și programare - Capacitatea de a utiliza noțiunile fundamentale ale programării. - Abilitatea de a elabora un program pentru o problemă practică.
Competențe transversale	- Abilitatea de a scrie o mini-documentație pentru o problemă dată. - Abilitatea de a scrie un algoritm și un program, precum și de a verifica corectitudinea acestora. - Insusirea de cunostinte de bază legate de algoritmică și elaborarea programelor.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	- Conceperea și realizarea unui program într-un limbaj de nivel înalt, precum și a documentației. - Alegerea unui algoritm/subalgoritm pentru o problema/subproblema data, studierea complexității, demonstrarea corectitudinii și verificarea și validarea acestuia, precum și întocmirea unei documentatii tehnice.
7.2 Obiectivele specifice	- Intelegerea notiunii de algoritm si invatarea limbajului Pseudocod pentru descrierea algoritmilor; - Formarea deprinderilor de proiectare a algoritmilor; - Cunoasterea unor algoritmi pentru unele clase de probleme: operatii cu vectori, matrice, polinoame, rezolvare de ecuatii si sisteme liniare, cautare, interclasare si sortare; - Formarea deprinderilor de concepere, executie, testare si punere la punct a programelor Pascal cu structurile de date simple; - Formarea unui stil de programare.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Algoritmi si descrierea lor - Notiunea de algoritm - Variabila, tip, specificare - Limbajul Pseudocod, structura liniara, alternativa, repetitiva	Expunerea, conversația, demonstrația didactică, modelarea, algoritmizarea, problematizarea.	
2. Subalgoritmi (Pseudocod) - Notiunea de subalgoritm - Parametrii formali - Definirea unui subalgoritm (functie si procedura) - Apelul unui subalgoritm - Importanta subalgoritmilor în programare	Descriere, explicatie, problematizarea, descoperirea, exemple.	
3. Codificarea algoritmilor Pseudocod în Pascal - Elemente de baza ale limbajului Pascal - Declaratii si Instructiuni - Structura unui program Pascal - Codificare	Expunerea, exemplificarea, studiul de caz.	

4. Recursivitate - Notiunea de Recursivitate - Recursivitate directa si indirecta - Exemple - Implementarea structuri de date statice și dinamice - Tablou, șir de caractere, mulțime, înregistrare - Liste, arbori	Expunerea, demonstrația didactică, algoritmizarea, problematizarea, descoperirea.	
5. Faze în viața unui program - Faze în realizarea unui program: specificare, proiectare, codificare, testare, documentare, verificare - Întreținerea programelor: corectiva, adaptiva, perfectiva - Testarea programelor: Metoda cutiei negre, Metoda cutiei transparente	Expunerea, conversația, demonstrația didactică, problematizarea, descoperirea.	
6. Corectitudinea algoritmilor - Stare în execuția unui program; - Calculul efectuat de un algoritm; - Rezultatul calculului efectuat de un algoritm; - Partial corectitudine, terminare, (total) corectitudine; - Metoda lui Floyd de demonstrare a corectitudinii	Expunerea, exemplificarea, demonstrația didactică, problematizarea, descoperirea.	
7. Dezvoltarea corectă a algoritmilor din specificații - Algoritm abstract - Ideea de rafinare - Reguli de rafinare - Exemple	Exemplificarea, studiul de caz.	
8. Metode generale de elaborare a algoritmilor - Programare modulară - Programare structurată - Teorema lui Bohm-Jacopini - Importanța clarității în programare - Elemente de claritate: indentare, spațiere, denumiri, ... - Stil în programare - Metoda top-down și Rafinarea în pași succesivi - Metoda top-down - Metoda bottom-up - Rafinarea în pași succesivi	Expunerea, conversația, algoritmizarea, problematizarea, descoperirea, exemplificarea.	
9. Tipuri Abstracte de Date - Specificarea unui TAD - Unit Pascal - Implementarea unui TAD în Pascal - Folosirea unui TAD în programare	Expunerea, conversația, modelarea, algoritmizarea, problematizarea, exemplificarea.	
10. Metoda Greedy - Prezentarea generală a metodei Greedy - Algoritm Greedy - Exemple și contraexemple - Euristica Greedy - Metoda Divide et Impera - Prezentare generală - Descrierea subalgoritmului - Exemple	Expunerea, conversația, demonstrația didactică, modelarea, algoritmizarea, problematizarea, descoperirea, exemplificarea, alte metode.	
11. Metoda Backtracking - Prezentarea generală a metodei Backtracking - Algoritm (subalgoritm) Backtracking - Extinderi ale metodei Backtracking - Exemple	Expunerea, modelarea, algoritmizarea, problematizarea, descoperirea, exemplificarea.	

12. Metode euristice - Strategii euristice, aproximarea solutiei - Exemple Alte metode - Metoda programarii dinamice - Metoda Branch and Bound	Expunerea, conversația, demonstrația didactică, modelarea, algoritmizarea, problematizarea, descoperirea, exemplificarea.	
13. Complexitatea algoritmilor - Definitia complexitatii - Complexitatea ca timp de executie - Complexitatea ca memorie suplimentara necesara - Analiza empirica si analiza asimptotica - Notatia asimptotica: o-mare, o-mic, omega-mare, omega-mic, theta - Proprietati - Ordine de marime particulare - Compararea algoritmilor din punctul de vedere al eficientei - Complexitatea structurala	Expunerea, descrierea, demonstrația didactică, explicația, studii de caz.	
14. Algoritmi de cautare si complexitatea lor - Specificarea problemei de cautare - Metode de cautare: parcurgere secventiala, cautare binara - Complexitatea algoritmilor de cautare Algoritmi de sortare si complexitatea lor - Specificarea problemei de sortare - Metode de sortare - Metoda bulelor (BubbleSort) - Sortare bazata pe selectarea minimului/maximului (SelectSort) - Sortare prin inserare (InsertSort) - Sortare rapida (QuickSort) - Sortare prin interclasare (MergeSort) - Complexitatea algoritmilor de sortare	Expunerea, descrierea, demonstrația didactică, explicația, studii de caz.	

Bibliografie

1. Frentiu, M., H.F. Pop, Fundamentals of Programming, Cluj University Press, 2006, 220 pagini
2. M.Frentiu si altii, Manualul incepatorului in Programarea Pascal, Ed. Microinformatica, Cluj-Napoca, 1995, 252 pagini, ISBN 973-9215-04-1
3. M.Frentiu si altii, Programare Pascal. Programe ilustrative, probleme propuse, pentru elevi si studenti, Ed. Promedia, 1995, 229 pagini, ISBN 973-96862-1-4.
4. M.Frentiu, V.Prejmereanu, Algoritmica si programare, Lito. Univ. Babes-Bolyai, Cluj-Napoca, 1995, 261 pagini.
5. Frentiu, M., I.Lazar, S.Motogna si V.Prejmereanu, vol.I - Elaborarea algoritmilor, 1997, 188 pagini; vol.II - Programare Pascal, 392 pagini; Ed.Universitatii Babes-Bolyai, Cluj-Napoca.
6. M.Frentiu si B.Parv, Elaborarea programelor. Metode si tehnici moderne, Ed.Promedia, Cluj-Napoca, 1994, 217 pag.
7. Kasa Z., Ismerkedes az informatikaval, Editura Dacia, 1983.
8. Kasa, Z., Algoritmusok tervezese, Ed. Studium, Cluj, 1994.
9. Knuth, D., Tratat de programarea calculatoarelor, Ed. Tehnica (Algoritmi fundamentali; Cautare si sortare)
10. Livovschi, L., H.Georgescu, Sinteza si analiza algoritmilor, Editura St. si Enciclopedica, Bucuresti, 1986.

8.2 Seminar	Metode de predare	Observații
1. Algoritmi si descrierea lor - Algoritmi simplii (fără subprograme) - Algoritmi de bază (elementari) - Tablouri unidimensionale și bidimensionale	Conversația, demonstrația didactică, modelarea, algoritmizarea.	
2. Subalgoritmi (Pseudocod) - Descrieri de subalgoritmi usuali. - Corelarea parametrilor (apel/definire).	Conversația, algoritmizarea, problematizarea, descoperirea.	
3. Codificarea algoritmilor elemenatri Pseudocod în Pascal - Programe Pascal simple, - Descriere și apeluri de subprograme.	Conversația, dezbaterea, algoritmizarea, descoperirea.	
4. Programe și subprograme Pascal care utilizează - Recursivitatea - Structuri de date statice (tablou, șir de caractere, mulțime, înregistrare) - Liste, arbori	Conversația, modelarea, descoperirea, algoritmizarea.	
5. Structuri de date dinamice - Liste simple și duble (stivă, coadă, ...) - Arbori binari (creare, traversare, ...)	Conversația, modelarea, descoperirea, algoritmizarea.	
6. Testarea programelor (Metoda cutiei transparente) Corectitudinea algoritmilor (partiala, terminare)	Conversația, modelarea, descoperirea.	
7. Dezvoltarea corecta a algoritmilor din specificatii - Reguli de rafinare - Exemple de dezvoltare corectă	Conversația, descoperirea, evaluarea.	
8. Programare modulara și Programare structurata Metoda top-down si Rafinarea în pasi succesivi	Conversația, modelarea.	
9. Tipuri Abstracte de Date - Specificarea unui TAD - Implementarea unui TAD în Pascal și utilizarea lui	Conversația, modelarea, problematizarea, descoperirea, algoritmizarea.	
10. Metoda Greedy - Exemple si contraexemple - Implementări Metoda Divide et Impera - Exemple și implementări	Exemplificarea, conversația, modelarea, problematizarea, descoperirea, algoritmizarea.	
11. Metoda Backtracking - Exemple și implementări	Exemplificarea, conversația, modelarea, problematizarea, algoritmizarea.	
12. Metode euristice - Exemple și implementări Alte metode - Metoda programarii dinamice	Exemplificarea, conversația, modelarea, problematizarea, descoperirea, algoritmizarea.	
13. Complexitatea algoritmilor de cautare - efectuarea calculelelor de complexitate pentru algoritmi uzuali	Dezbaterea, descoperirea.	
14. Complexitatea algoritmilor de sortare - efectuarea calculelelor de complexitate pentru algoritmi uzuali Lucrare de control	Dezbaterea, descoperirea, evaluarea.	

Bibliografie		
1. M.Frențiu, I.Lazăr, <i>Bazele Programării: Proiectarea Algoritmilor</i> , 2000, Ed. Univ. Petru Maior, Tg.Mureș 184 pagini ISBN 973-8084-06-7 2. M.Frentiu, <i>Verificarea Corectitudinii Programelor</i> , Ed.Univ."Petru-Maior", Tg.-Mureș, 2001, 116 pagini, ISBN 973-8084-32-6 3. F.Boian, M.Frențiu, I.Lazăr și L.Tambulea, <i>Informatica de bază</i> , Ed.Universității "Babeș-Bolyai", Cluj-Napoca, 2005, 233 pagini, ISBN 973-9261-18-3. 4. M.Frentiu, <i>Verificarea și Validarea Programelor</i> , Ed. Presa Universitară clujană, 2007, 173 pagini, ISBN 978-973-610-684-7.-2. 5. Knuth, D.E., <i>Tratat de programare a calculatoarelor. Algoritmi fundamentali</i> . Ed.Tehnica, Bucuresti 1974 6. Knuth, D.E., <i>Tratat de programare a calculatoarelor. Sortare si cautare</i> . Ed.Tehnica, Bucuresti 1976		
8.3 Laborator	Metode de predare	Observații
1. Structura documentației (lucrării/temei de laborator) Descrierea unui algoritm.	Explicația și justificarea.	
2. Descrierea de subalgoritmi și apelul acestora	Explicația și justificarea.	
3. Realizarea unui program simplu în limbajul Pascal	Dialogul, exemplificarea, dezbateră, evaluarea.	
4. Scrierea de subprograme și apelul lor dintr-un program.	Dialogul, exemplificarea, dezbateră.	
5. Subprograme recursive apelate dintr-un program care folosește vectori și matrice (tablouri uni/bi dimensionale).	Dialogul, exemplificarea, explicația.	
6. Programe care utilizează alte structuri de date statice - Inregistrări, mulțimi, string-uri, ...	Dialogul, exemplificarea, explicația, evaluarea.	
7. Programe care utilizează structuri de date dinamice - Liste - Arbori binari	Dialogul, exemplificarea, explicația.	
8. Testarea programelor: Metoda cutiei negre, Metoda cutiei transparente	Dialogul, exemplificarea, explicația.	
9. Demonstrarea corectitudinii unui algoritm. - Partial corectitudine, terminare, (total) corectitudine;	Explicația justificarea, evaluarea.	
10. Scrierea de programe utilizând - Metoda top-down, - Rafinarea în pași succesivi	Exemplificarea, dialogul, dezbateră.	
11. Tipuri Abstracte de Date - Specificarea unui TAD - Implementarea unui TAD în Pascal (Scrierea unui modul) - Utilizarea TAD-ului descris	Dialogul, exemplificarea, explicația, dezbateră, evaluarea.	
12. Metoda Greedy - Program de rezolvare a unei probleme utilizând această metodă Metoda Divide et Impera - Program de rezolvare a unei probleme utilizând această metodă	Dialogul, exemplificarea, explicația.	
13. Metoda Backtracking - Program de rezolvare a unei probleme utilizând această metodă Metode euristice - Program de rezolvare a unei probleme utilizând această tehnică	Dialogul, exemplificarea, explicația.	
14. Proba practică, evaluare finală a activității de laborator.	Evaluarea.	

Bibliografie

1. M.Frențiu, I.Lazăr, *Bazele Programării: Proiectarea Algoritmilor*, 2000, Ed. Univ. Petru Maior, Tg.Mureș 184 pagini ISBN 973-8084-06-7
2. M.Frentiu, *Verificarea Corectitudinii Programelor*, Ed.Univ."Petru-Maior", Tg.-Mureș, 2001, 116 pagini, ISBN 973-8084-32-6
3. F.Boian, M.Frențiu, I.Lazăr și L.Tambulea, *Informatica de bază*, Ed.Universității "Babeș-Bolyai", Cluj-Napoca, 2005, 233 pagini, ISBN 973-9261-18-3.
4. M.Frentiu, H.F.Pop, *Fundamentals of Programming*, Ed. Presa Universitară clujană, 2006, 220 pagini, ISBN (10)973-610-453-2 și (13)978-973-610-453-4.
5. M.Frentiu, H.F.Pop, G.Serban, *Programming Fundamentals*, Ed. Presa Universitară clujană, 2007, 234 pagini, ISBN (10)973-610-494-X și (13)978-973-610-494-7.
6. M.Frentiu, *Verificarea și Validarea Programelor*, Ed. Presa Universitară clujană, 2007, 173 pagini, ISBN 978-973-610-684-7.-2.

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Cursuri similare există în toate facultățile de informatică atât din țară cât și din străinătate;
- Fundamentele programării este un curs de bazele informaticii care vizează noțiunile de algoritmică și de programare fundamentală.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Aplicarea noțiunilor teoretice în realizarea programelor	Examen scris	50%
	Rezolvarea problemelor	Examen practic	15%
10.5 Seminar / laborator	Scrierea de algoritmi si programe	Lucrare de control (sem.)	15%
	Documentatie si Programe realizate	Documentatia realizată Programe predate	10% 10%
10.6 Standard minim de performanță			
• Nota finală se calculează conform celor cinci ponderi de mai sus, cu conditia ca toate notele să fie cel puțin 5.			

Data completării

15 mai 2013

Semnătura titularului de curs

Conf. Dr. Diosan Laura

Semnătura titularului de seminar

Conf. Dr. Diosan Laura

Data avizării în departament

.....

Semnătura directorului de departament

Prof. Dr. Parv Bazil