

**UNIVERSITATEA DIN BACĂU**

FACULTATEA \_\_\_\_DE ȘTIINȚE\_\_\_\_\_

DEPARTAMENTUL \_\_\_\_\_

CATEDRA \_\_MATEMATICĂ-INFORMATICĂ\_\_\_\_\_

Domeniul \_\_\_\_\_

Specializarea \_\_\_\_\_

**PROGRAMA ANALITICĂ A DISCIPLINEI:****\_\_FUNDAMENTELE PROGRAMĂRII\_\_**

Cod disciplină.....

An studiu	Semestrul	Durata (săptămâni)	Curs	Seminar	Laborator	Proiect	Credite	Total ore semestru	
			Ore săptămânal					Total ore	Studiu individual
al III-lea	I	14	2	1					

## Statut disciplină (se marchează cu X)

Obligatorie	Opțională	Facultativă
x		

## Categorie disciplină (se marchează cu X)

Fundamentală	În domeniu	De specialitate	Complementară
			x

Discipline	Obligatorii	Algoritmi si Structuri de Date
anterioare	Recomandate	Arhitectura Sistemelor de Calcul, Programarea Functionala

**Obiectivele disciplinei în termeni de competențe specifice (curs și aplicații)**

Asigurarea competențelor necesare viitorilor profesori de matematică (matematică – informatică) pentru a putea completa normele cu ore de informatică și limbaje de programare. Asigură matematicienilor posibilitatea de a realiza aplicații informatice specifice în limbajul funcțional Haskell, limbaj care se aseamănă parțial cu matematica. Asigură cunoașterea paradigmatelor programării, la nivel de bază.

(maxim 5 rânduri în corelare cu obiectivele și misiunea specializării)

**CONȚINUTUL DISCIPLINEI**

**Curs** (7 capitole a câte două cursuri) (capitole/subcapitole, număr de ore, maxim 1 pagină)

1. Programarea imperativă structurată, limbajul pseudocod, felul cum se predă un limbaj imperativ. Noțiunile fundamentale privind limbajele imperative structurate. Compararea dintre paradigmele programării, programarea imperativă - specific. Limbajul pseudocod Rodin.
2. Programarea funcțională, limbaje funcționale, felul cum se predă un limbaj funcțional ca aplicație la orele de matematică. Noțiunile fundamentale privind limbajele funcționale.

Comparația dintre paradigmele programării, programarea funcțională - specificul ei. Limbajul funcțional Haskell – introducere prin exemple.

3. Programarea logică, limbajul Prolog. Clauze, premise (ipoteze) sintaxă, motorul de inferențe, backtracking. Noțiunile fundamentale privind limbajul Prolog și logica matematică. Comparația dintre paradigmele programării, programarea logică - specificul ei. Limbajul Prolog. – introducere prin exemple.
4. Programarea orientată obiect cu șabloane. Obiecte, date, metode, șabloane. Noțiunile fundamentale privind Programarea orientată obiect și aspecte matematice. Comparația dintre paradigmele programării, programarea orientată obiect - specificul ei. Limbajul C++ (sau alt limbaj) – introducere prin exemple.
5. Algoritmi, structuri de date și tehnici de programare esențiale. Iterativitate vs. recursivitate. Liste, arbori. Folosirea funcțiilor de nivel superior (funcțiilor de funcții) pentru a scrie programe scurte.
6. Noțiuni de matematică reprezentate / puse la dispoziție de limbajele de programare funcțională. Mulțimi definite descriptiv, (comprehensiuni de liste), mulțimi numărabile infinite, arbori, grafuri, liste, funcții, compunere și aplicare parțială de funcții, domenii.
7. Noțiuni avansate de algebră. Mulțimi definite inductiv și funcții definite pe ele. Elemente de teoria categoriilor care stau la baza construcției limbajelor de programare. (Introducere).  
Recapitulare.

#### **Seminar** (dacă este cazul)

Demonstrații de corectitudine pentru programe mici și includerea demonstrațiilor ca teste.  
Scrierea de programe de uz didactic pentru a se ilustra noțiunile predate.

#### **Lucrări de laborator** (dacă este cazul)

Aceleași teme cu exemple de programe, constituindu-se un portofoliu salvat pe un suport de date.

#### **Tematică proiect** (dacă este cazul)

## **BIBLIOGRAFIA**

1. Dan Popa, PASCALUL mileniului al III-lea, Programarea calculatoarelor în Oberon-2, noul limbaj al profesorului N.Wirth, Edusoft 2005
2. Introducere în Haskell 98 prin exemple, Edusoft, 2007
3. Pagina Grupului Român de utilizatori de Haskell Ro/Haskell  
<http://www.haskell.org/haskellwiki/Ro/Haskell>
4. Dan Popa – Programarea orientata obiect – Universitatea Bacau, 2005,2006,2007 e-book de uz intern folosit pentru IFR
5. Mandriva Linux Team – Manualele de GNU Prolog incluse în distribuție, Mandriva Linux, Paris, 2008,2009 (eventual și alte manuale incluse în distribuție)

**STABILIREA NOTEI FINALE LA FUNDAMENTELE PROGRAMĂRII**

Forma de verificare (Examen, Colocviu, Verificare pe parcurs)		Examen
Modalitatea de susținere (Scris și Oral, Oral, Test grilă, etc.)		Scris și oral
<b>NOT ARE</b>	Răspunsuri la examene, colocviu	(10)100%
	Evaluare activități aplicative (laborator, proiect)	bonus
	Prezență activă la curs și seminar	bonus
	Lucrare de verificare	neașteptată
	Teme de casă	bonus
	Alte activități (specificați)	
	<b>TOTAL PUNCTE SAU PROCENTE</b>	<b>10 (100%)</b>

Condiții minime de promovare (cum se obține nota 5)	Condiții de obținere a notei maxime
răspuns + bonus mai mare sau egal cu 4.50	răspuns + bonus mai mare sau egal cu 9.50

Forme activitate	Metode didactice folosite
Curs	Expunere frontală cu exemple problematizate
Seminar	Exerciții
Laborator	
Proiect	

## Titular disciplină

Numele și Prenumele	Popa Dan
Instituția	Universitatea Bacău
Departament / Catedră	<i>MATEMATICĂ-INFORMATICĂ</i>
Titlul științific	Drd.
Gradul didactic	Lector
Încadrarea (normă de bază în Univ./asociat)	normă de bază
Vârsta	38
Semnătura	

	Funcția	Grad didactic	Prenumele și Numele	Semnătura	Data
Avizat*					
Aprobat**					

\* Avizează șeful catedrei care coordonează specializarea

\*\* Aprobă șeful catedrei care are cursul în statul de funcțiuni