

FIȘA DISCIPLINEI
Paradigme de programare
UP.03.D.4.A.03.41
Anul universitar 2022-2023

1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Electronică, Comunicații și Calculatoare
1.3	Departamentul	Electronică, Calculatoare și Inginerie Electrică
1.4	Domeniul de studii	Calculatoare și tehnologia informației
1.5	Ciclul de studii	Licență
1.6	Programul de studii / Calificarea	Calculatoare / Programator de sistem informatic (251204)

2. Date despre disciplină

2.1	Denumirea disciplinei					Paradigme de programare					
2.2	Titularul activităților de curs					Conf.univ.dr. Doru CONSTANTIN					
2.3	Titularul activităților de laborator					Conf.univ.dr. Doru CONSTANTIN					
2.4	Anul de studii	II	2.5	Semestrul	II	2.6	Tipul de evaluare	E	2.7	Regimul disciplinei	A

3. Timpul total estimat

3.1	Număr de ore pe săptămână	3	3.2	din care curs	2	3.3	laborator	1
3.4	Total ore din planul de învățământ	42	3.5	din care curs	28	3.6	laborator	14
Distribuția fondului de timp								ore
Studiul după manual, suport de curs, bibliografie și notițe								10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								10
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								9
Tutorat								2
Examinări								2
Alte activități								-
3.7	Total ore studiu individual	33						
3.8	Total ore pe semestru	75						
3.9	Număr de credite	3						

4. Precondiții (acolo unde este cazul)

4.1	De curriculum	Parcursirea și/sau promovarea următoarelor discipline: structuri de date, programarea calculatoarelor, analiza algoritmilor
4.2	De competențe	abilități de sintetizare și raționament logic

5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sală cu o capacitate de minim 70 locuri dotată cu videoproiector și ecran de proiecție, 1 tablă.
5.2	De desfășurare a laboratorului/proiectului	Sala de laborator dotată cu calculatoare, capacitate maximă 18 studenți/laborator.

6. Competențe specifice acumulate

Competențe profesionale	C4. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații C5. Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software C6. Proiectarea sistemelor inteligente C6.1 Descrierea componentelor sistemelor inteligente; C6.2 Utilizarea de instrumente specifice domeniului pentru explicarea funcționării sistemelor inteligente; C6.3 Aplicarea principiilor și metodelor de baza pentru specificarea de soluții la probleme tipice utilizând sisteme inteligente; C6.5 Dezvoltarea și implementarea de proiecte profesionale pentru sisteme inteligente.
Competențe transversale	CT2. Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	- Cunoașterea modului în care principalele concepte aferente programării sunt materializate și utilizate în raport cu paradigma de programare aleasă. - Redarea comparativă a elementelor esențiale ale unor limbaje de programare și aplicarea unor tehnici de programare specifice limbajelor studiate.
7.2 Obiectivele specifice	<i>Obiective cognitive:</i> ■ Cunoașterea principiilor ce caracterizează modelele de calculabilitate, a diverselor perspective conceptuale asupra noțiunii de calculabilitate efectivă; ■ Cunoașterea și înțelegerea conceptelor și a elementelor de paradigme de programare și analiza influenței perspectivei alese asupra procesului de modelare și rezolvare a problemelor. <i>Obiective procedurale:</i> ■ Formarea deprinderilor și abilitatea de a utiliza un mediu de programare; ■ Formarea deprinderilor și abilitatea de a utiliza elementele unui limbaj de programare, cunoașterea

mecanismelor expresive aferente paradigmelor, cu accent pe aspectul comparativ.
■ Obiective atitudinale: Rigurozitate în proiectarea și implementarea algoritmilor analizați.

8. Conținuturi

8.1. Curs		Nr. de ore	Metode de predare	Observații Resurse folosite
1	Introducere în paradigme: similarități și diferențe dintre principalele paradigme de programare	2	Prelegere Dezbateri Explicația Descrierea Exemplificarea Problematizarea Exercițiul	Calculator, Videoproector
2	Paradigma imperativă: programare structurată versus programare nestructurată	2		
3	Paradigma imperativă: programare orientată spre obiect (C++, JAVA, C#, Python, Eiffel)	2		
4	Modelare orientată spre automate (automata based programming: Finite State Machines, Time automata, Abstract State Machine Language-AsmL)	2		
5	Paradigma declarativă: programare funcțională	2		
6	Paradigma declarativă: programare logică	2		
7	Paradigma orientată spre evenimente (Java, C#)	2		
8	Programare orientată spre aspecte (AOP)	2		
9	Paradigma secvențială versus concurența (Erlang)	2		
10	Programare paralelă orientată spre procese	2		
11	Paradigma programării orientate spre agenți (agent oriented: JADE)	2		
12	Programarea sistemelor distribuite (IDL, CORBA, DISPY)	2		
13	Metaprogramare reflectivă orientată spre atribute	2		
14	Metaprogramare orientată spre șabloane	2		

Bibliografie

1. Note de curs și laborator – suport electronic – Tudor Bălănescu.
2. E. Diaconescu: Limbajul Erlang. Programarea concurență a sistemelor concurențe și distribuite. Editura Alabastră, Cluj, 2012.
3. L. State: Introducere în programarea logică, Editura Fundației România de mâine, București, 2004.
4. David Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems Hardcover, Addison-Wesley Professional, 2002.
5. I. Bratko: PROLOG, Addison Wesley, 2001.
6. J. Hughes, S.P. Jones, P. Wadler, "A history of Haskell: being lazy with class", ACM SIGPLAN Conf. on History of Programming Languages. San Diego, California, 2007; L. Sterling, Shapiro: The Art of Prolog, MIT Press, 2001.
7. Crimi, C., A. Guercio, G. Pacini, G. Tortora, and M. Tucci, "Automating Visual Language Generation", IEEE Transactions on Software Engineering, vol. 16, no. 10, pp. 1122-1135, 1990.
8. Margaret M. Burnett, Herkimer J. Gottfried, Graphical definitions: expanding spreadsheet languages through direct manipulation and gestures. ACM Trans. Comput.-Hum. Interact. 5, 1, 1998, 1-33.

8.2. Aplicații – Laborator		Nr. de ore	Metode de predare	Observații Resurse folosite
1	Aplicații introductive privind paradigmele de programare	1	Studiul de caz Lucrul în grup Problematizarea Exercițiul Dezbateri	Calculator, Videoproector
2	Aplicații privind paradigma structurată	1		
3	Aplicații privind paradigma orientată obiect (CPP, JAVA, C#, Python)	2		
4	Aplicații privind reprezentările abstracte (UML, FSM, Time Automata)	1		
5	Aplicații de programare funcțională	1		
6	Aplicații privind paradigma orientată eveniment (Java, C#)	2		
7	Aplicații privind modelele de proiectare (Java)	2		
8	Aplicații privind paradigma secvențială versus concurența (Java, Erlang)	1		
9	Aplicații de programare logică	1		
10	Aplicații privind paradigma programării vizuale	1		
11	Aplicații de programare orientată spre aspecte	1		

Bibliografie

1. Note de curs și laborator – suport electronic – Tudor Bălănescu.
2. L. State: *Introducere în programarea logică*, Editura Fundației România de mâine, București, 2004.
3. David Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems Hardcover, Addison-Wesley Professional, 2002.
4. I. Bratko: *PROLOG*, Addison Wesley, 2001.
5. J. Hughes, S.P. Jones, P. Wadler, "A history of Haskell: being lazy with class", ACM SIGPLAN Conf. on History of Programming Languages. San Diego, California, 2007; L. Sterling, Shapiro: The Art of Prolog, MIT Press, 2001.
6. Crimi, C., A. Guercio, G. Pacini, G. Tortora, and M. Tucci, "Automating Visual Language Generation", IEEE Transactions on Software Engineering, vol. 16, no. 10, pp. 1122-1135, 1990.
7. Margaret M. Burnett, Herkimer J. Gottfried, Graphical definitions: expanding spreadsheet languages through direct manipulation and gestures. ACM Trans. Comput.-Hum. Interact. 5, 1, 1998, 1-33.

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

Coroborarea conținuturilor disciplinei cu așteptările comunității epistemice se realizează prin stabilirea de întâlniri cu principalii actori de pe piața IT locală (RoWeb, Lisa, Prodinf, Kepler, Osf, Endava, etc.), precum și prin vizite la firmele de profil și schimburi de bune practici cu cadre didactice din alte universități.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	a) examen	a) scris - verificare cunoștințe teoretice și a elementelor de implementare	50%
10.5 Laborator	a) activitate laborator și teme de casă b) rezolvarea problemelor folosind cunoștințele cumulate la laborator și curs	a) evaluarea activității și a temelor b) evaluare periodică a implementărilor realizate	10% 40%
10.6 Standard minim de performanță	*Notă minimă 5 la activitățile de laborator și notă minimă 5 la fiecare din subiectele de la examenul final. *Set de cunoștințe minimale pentru promovarea examenului final: cunoașterea principiilor de bază privind paradigmele de programare.		

Data completării
12.09.2022

Titular de curs și proiect
Conf.univ.dr. Doru CONSTANTIN

Titular de laborator
Conf.univ.dr. Doru CONSTANTIN

Data avizării în Departament
15.09.2022

Director Departament (prestator)
Conf.univ.dr. Doru CONSTANTIN

Director Departament (beneficiar)
Prof.univ.dr.ing. Gheorghe ȘERBAN