

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare logică și funcțională						
2.2 Titularul activităților de curs	Prof. dr. CZIBULA Gabriela						
2.3 Titularul activităților de seminar	Prof. dr. CZIBULA Gabriela						
2.4 Anul de studiu	2	2.5 Semestrul	3	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	3	Din care: 3.2 curs	2	3.3 seminar/laborator	1 lab
3.4 Total ore din planul de învățământ	42	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					17
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					30
Tutoriat					18
Examinări					18
Alte activități:					-
3.7 Total ore studiu individual	108				
3.8 Total ore pe semestru	150				
3.9 Numărul de credite	6				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de competențe	

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> Sală de curs cu videoproiector
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Sală de laborator cu calculatoare dotate cu limbajele de programare CLisp și SWIProlog

6. Competențele specifice acumulate

Competențe profesionale	<p>C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</p> <p>C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date</p> <p>C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente</p>
Competențe transversale	<p>CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> Să deprindă studentul cu paradigma programării declarative (programarea funcțională și programarea logică).
7.2 Obiectivele specifice	<ul style="list-style-type: none"> Să introducă câte un limbaj de programare pentru fiecare din aceste paradigme (Common Lisp și Prolog). Să inducă ideea utilizării acestor paradigme în funcție de necesitățile aplicațiilor. Să asigure baza necesară urmării unor cursuri avansate de programare declarativă.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
<i>Programare Logică. Limbajul PROLOG</i>		
1. Programare și limbaje de programare. Programare imperativă vs. programare declarativă. Introducere. Recursivitate. Exemple.	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
2. Elemente fundamentale ale limbajului Prolog. Fapte și reguli Prolog. Întrebări. Strategia de control în Prolog. Variabile și propoziții compuse. Variabile anonime. Reguli de definire a potrivirilor. Model de flux. Secțiunile unui program Prolog. Exemple.	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
3. Programul Prolog. Domenii predefinite. Întrebări interne și externe. Predicate cu aritate multiplă. Simbolul IF (Prolog) și instrucțiunea IF (alte limbaje). Directive de compilare. Expresii aritmetice și	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	

comparații. Operații de intrare / ieșire. Șiruri de caractere.		
4. Backtracking. Controlarea backtracking-ului. Predicatele fail și ! (cut). Utilizarea lui !. Tipuri de tăieturi. Predicatul “not”. Liste Prolog. Recursivitate. Exemple de tratare a backtracking-ului. Găsirea tuturor soluțiilor în același timp. Exemple de predicate Prolog. Predicate nedeterministe	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
5. Obiecte compuse și funcții. Unificarea obiectelor compuse. Argumente de tipuri multiple; liste eterogene. Compararea obiectelor compuse. Backtracking cu ciclări. Exemple de proceduri recursive. Cadrul stivei. Optimizarea prin recursivitate de coadă. Utilizarea tăieturii pentru păstrarea recursivității de coadă.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
6. Structuri de date recursive. Arborii ca structuri de date. Construirea și traversarea unui arbore. Arbori de căutare.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<i>Programare Funcțională. Limbajul LISP</i>		
7. Importanța programării funcționale ca noua metodologie de programare. Istoric și prezentare a limbajului LISP. Elemente de bază Lisp. Structuri dinamice de date. Reguli sintactice și semantice. Clasificarea funcțiilor Lisp. Funcții primitive în Lisp.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
8. Predicate de bază în Lisp. Predicate pentru liste; pentru numere. Funcții logice și aritmetice. Definirea funcțiilor utilizator. Ramificarea prelucrărilor. Metoda variabilei colectoare. Exemple.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
9. Gestiunea simbolurilor. Alte funcții de acces la liste. OBLIST și ALIST. Funcții cu caracter destructiv. Comparații. Alte funcții interesante. Exemple.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
10. Mecanisme definiționale evoluate Forma EVAL. Forme funcționale; funcțiile FUNCALL și APPLY. Expresii LAMBDA. Expresii LABEL. Exemple	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
11. Generatori, argumente funcționale. Funcții MAP. Forme iterative. Exemple.	<ul style="list-style-type: none"> • 	
12. Alte elemente ale limbajului Lisp. Structuri de date. Macrodefiniții. Argumente opționale. Exemple.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
13. Examen scris PROLOG+LISP	Lucrarea scrisă	

Bibliografie

1. CZIBULA G., POP H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastra, Cluj-Napoca, 2012
2. POP H.F., SERBAN G., Programare în Inteligența Artificială - Lisp și Prolog, Editura Albastra, Cluj-Napoca, 2003
3. <http://www.ifcomputer.com/PrologCourse>, Lecture on Prolog
4. <http://www.lpa.co.uk>, Logic Programming
5. FIELD A., Functional Programming, Addison Wesley, New York, 1988.
6. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.

8.2 Laborator	Metode de predare	Observații
		Laboratorul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
Lab 1: Recursivitate (Pseudocod) (lucrare scrisă, notată)	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • Se predă tema primită la Lab 1 • Se primește tema pentru Lab 2
Lab 2: Liste în Prolog	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • Se predă tema primită la Lab 2 • Se primește tema pentru Lab 3
Lab 3: Test practic Prolog (1)	<ul style="list-style-type: none"> • Test practic 	1 oră
Lab 3: Arbori în Prolog. Gestiunea listelor în Prolog.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • Se predă tema primită la Lab 3 • Se primește tema pentru Lab 4
Lab 4: Backtracking în Prolog	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • 1 oră • Se predă tema primită la Lab 4 • Se primește tema pentru Lab 5
Lab 4: Test practic Prolog (2)	<ul style="list-style-type: none"> ▪ Test practic 	1 oră
Lab 5: Programare recursivă în Lisp	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • Se predă tema primită la Lab 5 • Se primește tema pentru Lab 6
Lab 6: Test practic Lisp (1)	<ul style="list-style-type: none"> • Test practic 	1 oră
Lab 6: Folosirea funcțiilor MAP.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • Se predă tema primită la Lab 6 • Se primește tema pentru Lab 7
Lab 7: Programare iterativă în Lisp	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	<ul style="list-style-type: none"> • 1 oră • Se predă tema primită la Lab 7
Lab 7: Test practic Lisp (2)	<ul style="list-style-type: none"> • Test practic 	1 oră

Bibliografie

1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012
2. Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp.
3. Documentatia produselor: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog.
4. <http://www.swi-prolog.org/>

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

- Conținutul disciplinei este în concordanță cu ceea ce se face în alte centre universitare din țară și din străinătate.
- Conținutul disciplinei asigură cunoștințele fundamentale necesare pentru programare în Lisp și Prolog la eventualii angajatori.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<ul style="list-style-type: none"> • Corectitudinea și completitudinea cunoștințelor asimilate. • Gradul de asimilare a limbajului de specialitate. 	Evaluare scrisă: examen scris Prolog+Lisp (curs 14)	60%
10.5 Seminar/Laborator	<ul style="list-style-type: none"> • Implementarea în Lisp și Prolog a conceptelor și algoritmilor prezentați la curs • Redactarea documentației de laborator • Respectarea termenelor de predare. 	Documentații și programe	10%
		Lucrare de laborator - recursivitate	6%
		Test practic Prolog	6%
		Test practic Prolog	6%
		Test practic Lisp	6%
		Test practic Lisp	6%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Cunoașterea elementelor fundamentale de teorie, rezolvarea unei aplicații simple într-un limbaj declarativ. Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaștere și înțelegere a domeniului, că este capabil să exprime cunoștințele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni și de a utiliza cunoștințele în rezolvarea unor probleme. • Pentru promovare sunt necesare următoarele criterii minimale: nota minim 4 la fiecare din lucrările scrise, predarea a minim 5 lucrări de laborator și nota finală minim 5. 			

Data completării

30.04.2014

Semnătura titularului de curs

Prof. dr. Gabriela Czibula

Semnătura titularului de seminar

Prof. dr. Gabriela Czibula

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Bazil Pârv