FIŞA DISCIPLINEI[1]

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea „Politehnica” din Timişoara
1.2 Facultatea[2] / Departamentul[3]	Automatică şi Calculatoare / Automatică și Informatică Aplicată
1.3 Catedra	-
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licenţă
1.6 Programul de studii / Calificarea	Informatică/Informatician

2. Date despre disciplină

2.1 Denumirea disciplinei			**Programare Java**				
2.2 Titularul activităţilor de curs			Prof. dr. ing. Horia CIOCÂRLIE				
2.3 Titularul activităţilor de seminar			Asist. dr. ing. Oana Căuș				
2.4 Anul de studiu	2	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Opțională

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

3.1 Număr de ore pe săptămână	4	din care:3.2 curs (SI)	2	3.3 seminar/laborator (AA)	2
3.4 Total ore din planul de învăţământ	56	din care:3.5 curs	28	3.6 seminar/laborator	28
Distribuţia fondului de timp					ore
Studiul după manual, suport de curs, bibliografie şi notiţe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri					14
Tutoriat					9
Examinări					3
Alte activităţi					

3.7 Total ore studiu individual	48
3.8 Total ore pe semestru	104
3.9 Numărul de credite	4

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Programarea Calculatoarelor, Tehnici de Programare, Programarea orientată pe obiecte
4.2 de competenţe	• Cunostinte de matematică elementară (la nivel de liceu) • Cunostinte de programare algoritmică • Cunostinte de programare orientată pe obiecte

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală medie, Materiale suport: laptop, proiector, tablă.
5.2 de desfăşurare a seminarului/laboratorului	• Laborator cu 17-25 calculatoare, tabla • Mediu de programare LISP, ML

6. Competenţe specifice acumulate

[1] Formularul corespunde Fişei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

[2] Se înscrie numele facultăţii care gestionează programul de studiu căruia îi aparţine disciplina;

[3] Se înscrie numele departamentului căruia i-a fost încredinţată susţinerea disciplinei şi de care aparţine titularul cursului;

Competenţe profesionale[4]	• Programarea în limbaje de nivel înalt • Dezvoltarea şi întreținerea aplicațiilor informatice. • Utilizarea bazelor teoretice ale informaticii si a modelelor formale. • Utilizarea instrumentelor informatice in context interdisciplinar.
Competenţe transversale	Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţil or de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice și de comunicare în limba română și într-o limbă de circulație internațională.

7. Obiectivele disciplinei (reieşind din grila competenţelor specifice acumulate)

7.1 Obiectivul general al disciplinei	• Studiul conceptelor fundamentale încorporate în limbajele de programare (LP) si dezvoltarea acestor concepte odată cu evoluția LP; • Dobândirea unei imagini de ansamblu asupra domeniului limbajelor de programare
7.2 Obiectivele specifice	• Sistematizarea LP; • Aprecierea calită□ilor si lipsurilor unui LP; • Învățarea usoară si sistematică a unui nou LP; • Utilizarea eficientă a oricărui LP; • Selectarea corectă a LP potrivite pentru o anumită aplicație; • Proiectarea unui nou LP sau a unui subset (extensie) a unuia existent. • Deprinderi practice pentru programarea funcțională

8. Conţinuturi

8.1 Curs	Număr de ore	Metode de predare
1. Introducere. Calitățile unui limbaj de programare. Clasificări 1.1 Limbajul de programare în cadrul procesului de dezvoltare software 1.2 Criterii pentru evaluarea unui limbaj de programare 1.3 Cele trei famiii de limbaje de programare: imperative, func□ionale, declarative 1.4 Programarea secven□ială si programarea concurentă 1.5 Scurt istoric al dezvoltării limbajelor de programare	2	Prelegere susţinută de prezentări PPT, conversaţii, explicaţii, exemplificări
2. Reprezentarea formală a limbajelor de programare 2.1 Sintaxa 2.2 Semantica	2	
3. Implementarea limbajelor de programare 3.1 Interpretarea 3.2 Translatarea 3.3 Considerente comparative 3.4 Detalii privind procesul de compilare	2	
4. Atributele entităților unui limbaj de programare 4.1 Domeniul variabilelor 4.2 Durata de existență a variabilei. Alocarea memoriei 4.3 Valoarea variabilei. Tipul variabilei	2	
5. Transmiterea datelor ca parametri 5.1 Transmiterea prin adresă, prin copiere, prin nume 5.2 Transmiterea datelor ca parametri, în diverse limbaje de programare 5.3 Transmiterea subprogramelor ca parametri 5.4 Subprograme generice	2	
6. Tipuri de date – prezentare generală 6.1 Tipuri predefinite. Tipuri definite de programator. 6.2 Tipuri scalare. Tipuri de date structurate 6.3 Tipul pointer 6.4 Compatibilitatea tipurilor 6.5 Sistemul de tipuri al limbajului Pascal, al limbajului C, al limbajului Ada, al limbajului Lisp 6.6 Comparație. Limbaje puternic tipizate	2	
7. Tipuri de date abstracte 7.1 Descrierea datelor abstracte 7.2 Date abstracte în câteva limbaje de programare moderne: trecere în	2	

[4] Aspectul competenţelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competenţele care sunt precizate în Registrul Naţional al Calificărilor din Învăţământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4, programul de studii de la pct. 1.6 din această fişă şi materia în cauză

revistă 7.3 Date abstracte în Modula 2, Ada, C++		
8. Limbaje orientate pe obiecte 8.1 Programarea orientată pe obiecte 8.2 Programarea orientată pe obiecte în limbajele Java, C#, Lisp	2	
9. Structuri de control în limbajele de programare 9.1 Structuri de control la nivel de instruc□iune 9.2 Subprograme 9.3 tratarea excep□iilor	2	
10. Bazele teoretice ale programării funcționale 10.1 Lambda calculus 10.2 Evaluarea lenesă 10.3 Funcții de ordin superior 10.4 Tipuri; polimorfism	2	

Bibliografie

1. Carlo Ghezzi, Mehdi Jarayeri, Programming Languages, John Wiley 1987.
2. Ellis Horowitz, Fundamentals of Programming Languages, Computer Science Press, 1984.
3. Horia Ciocârlie, Universul limbajelor de programare, Editura Eurostampa, 2011.

8.2 Seminar/laborator	Număr de ore	Metode de predare
1. Despre LISP. Introducere în LISP	2	Expunere lucrare de laborator, discuţii, rezolvare de interfețe în tematica lucrării de laborator.
2. Crearea de noi proceduri si realizarea evaluărilor condiționate	2	
3. Recursivitate	2	
4. Primitivele MAPCAR si APPLY. Cicluri	2	
5. Definirea procedurilor anonime cu LAMBDA	2	
6. Liste de asociații	2	
7. PRINT si READ	2	
9. Lucrul cu matrici	2	
10. Lucrul cu celule de memorie	2	
11. Introducere în ML. Tipuri de bază. Variabile. Funcții	2	
12. Potrivirea de sabloane. Tipuri complexe	2	
13. Tipuri recursive	2	
14. Recuperări	2	

Bibliografie

1. P.H. Winston, B.K.P. Hein, LISP, Second Edition, Addison-Wesley, 1984.
2. Ileana Streinu, LISP, Editura Stiinţifică si Enciclopedică, 1986.
3. Chris Reade, Elements of Functional Programming, Addison-Wesley, 1989.

9. Corelarea conţinutului disciplinei cu cerinţele specialiştilor din domeniu şi cu aşteptările angajatorilor reprezentativi

Cunostințele predate sunt importante pentru dobândirea unei imagini de ansamblu asupra domeniului limbajelor de programare. Se ating urmatoarele scopuri:

- Sistematizarea LP;
- Aprecierea calităților si lipsurilor unui LP;
- Învățarea usoară si sistematică a unui nou LP;
- Utilizarea eficientă a oricărui LP.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Subiecte teoretice si exerciții care vizează înțelegerea conceptelor, problematicii si aplicării rezultatelor teoretice	Examen scris	40%
	Subiecte aplicative constând în rezolvarea unor probleme în limbajele de programare funcționale LISP sau ML	Examen scris	25%
10.5 Seminar /laborator	Rezolvarea problemelor corespunzătore lucrărilor de laborator	Prezentarea rezolvărilor, răspunsuri la întrebări	20%
	Teme de casă	Prezentarea rezolvărilor, răspunsuri la întrebări	10%
	Prezenţa	Evidenţa prezenţei	5 %

10.6 Standard minim de performanţă (volumul de cunoştinţe minim necesar pentru promovarea disciplinei şi modul în care se verifică stăpânirea lui)
• Stăpânirea conceptelor fundamentale ale LP: variabilă, constantă, parametru, argument, tip de dată, programare orientată pe obiecte, programare funcțională. • Proiectarea, testarea si executarea unui program de complexitate medie în LISP sau ML

11. Compatibilitate internaţională

- Princeton University
- Ilinois Institute of Technology
- University of Southern California

Data completării	Semnătura titularului de curs	Semnătura titularilor de seminar
15.03.2015	Prof. dr. Ing. Horia CIOCÂRLIE ……....	Asist.dr.ing. Oana CĂUȘ …….....

Data avizării în departament	Semnătura directorului de departament
15.03.2015	Prof. dr. ing. Vladimir Ioan CREȚU …….…….……