

# FIȘA DISCIPLINEI<sup>1</sup>

## 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnică Timișoara
1.2 Facultatea <sup>2</sup> / Departamentul <sup>3</sup>	Facultatea de Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod <sup>4</sup> )	ȘTIINȚE INGINEREȘTI APLICATE / 270
1.5 Ciclul de studii	Licenta
1.6 Programul de studii (denumire/cod/calificarea)	INFORMATICĂ INDUSTRIALĂ / 50 / Inginer

## 2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă <sup>5</sup>	Ingineria sistemelor de programe / DD						
2.2 Titularul activităților de curs	Ș. L. Dr. Iordan Anca-Elena						
2.3 Titularul activităților aplicative <sup>6</sup>	Ș. L. Dr. Ing. Abrudean Cristian						
2.4 Anul de studii <sup>7</sup>	IV	2.5 Semestrul	II	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei <sup>8</sup>	DO

## 3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)<sup>9</sup>

3.1 Număr de ore asistate integral/săptămână	3,5 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	1,5
3.1* Număr total de ore asistate integral/sem.	49 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	21
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	3,92 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1,92
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1
3.7* Număr total de ore activități neasistate/semestru	55 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			27
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			14
3.8 Total ore/săptămână <sup>10</sup>	7,42				
3.8* Total ore/semestru	104				
3.9 Număr de credite	4				

## 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> <li>Programarea calculatoarelor 1, Programarea calculatoarelor 2, Proiectarea algoritmilor, Programarea calculatoarelor și limbaje de programare, Baze de date, Programare Java</li> </ul>
4.2 de competențe	<ul style="list-style-type: none"> <li></li> </ul>

<sup>1</sup> Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 și cerințelor Standardelor specifice ARACIS valabile începând cu 01.10.2017.

<sup>2</sup> Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina.

<sup>3</sup> Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

<sup>4</sup> Se înscrie codul prevăzut în HG nr.140/16.03.2017 sau în HG similare actualizate anual.

<sup>5</sup> Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină fundamentală (DF), disciplină de domeniu (DD), disciplină de specialitate (DS) sau disciplina complementară (DC).

<sup>6</sup> Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

<sup>7</sup> Anul de studii în care este prevăzută disciplina în planul de învățământ.

<sup>8</sup> Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI), disciplină opțională (DO) sau disciplină facultativă (Df).

<sup>9</sup> Numărul de ore de la rubricile 3.1\*, 3.2\*,...,3.8\* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,...., 3.8. Informațiile din rubricile 3.1, 3.4 și 3.7 sunt chei de verificare folosite de ARACIS sub forma: (3.1)+(3.4) ≥ 28 ore/săpt. și (3.8) ≤ 40 ore/săpt.

<sup>10</sup> Numărul total de ore / săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.7.

## 5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"><li>• Sală de curs echipată cu videoproiector și conexiune la Internet.</li><li>• Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise.</li></ul>
5.2 de desfășurare a activităților practice	<ul style="list-style-type: none"><li>• Sală de laborator echipată cu videoproiector și computere.</li><li>• Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise.</li></ul>

## 6. Competențe la formarea cărora contribuie disciplina

Competențe specifice	<p>C 2.</p> <p>C 2.1. Descrierea structurii și a modului de funcționare a sistemelor informatice în general;</p> <p>C 2.2. Explicarea rolului, funcționalității și utilității sistemelor informatice în general și a sistemelor de prelucrare și gestiune a datelor în domeniul specializării;</p> <p>C 2.3. Utilizarea componentelor software ale sistemelor informatice, folosind algoritmi, protocoale, limbaje, structuri de date;</p> <p>C 2.4. Aprecierea caracteristicilor și calității sistemelor informatice;</p> <p>C 2.5. Prelucrarea și gestionarea datelor utilizând sisteme informatice dedicate.</p> <p>C 6.</p> <p>C 6.1. Descrierea principiilor de bază privind achiziția și transmisia de date din proces;</p> <p>C 6.2. Explicarea rolului componentelor sistemelor de achiziție de date aferente unui sistem informatic destinat conducerii automate a proceselor industriale;</p> <p>C 6.3. Configurarea sistemelor de achiziție și transmisie de date aferente proceselor industriale;</p> <p>C 6.4. Utilizarea adecvată a metodelor de evaluare a performanțelor sistemelor informatice și de validare a datelor achiziționate din proces;</p> <p>C 6.5. Implementarea componentelor sistemelor informatice de achiziție de date.</p> <p>•</p>
Competențele profesionale în care se înscriu competențele specifice	<ul style="list-style-type: none"><li>• C 2. Utilizarea sistemelor informatice de prelucrare și gestiune a datelor.</li><li>• C 6. Configurarea, implementarea și folosirea sistemelor de achiziție de date.</li></ul>
Competențele transversale în care se înscriu competențele specifice	<ul style="list-style-type: none"><li>•</li></ul>

## 7. Obiectivele disciplinei (asociate competențelor de la punctul 6)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"><li>• Scopul acestei discipline îl reprezintă însușirea principalelor concepte, metode și tehnici ale ingineriei sistemelor de programare, cu accent pe reutilizarea codului sursă atât din punct de vedere al programării orientate pe obiect, cât și din punct de vedere al programării generice în limbajul de programare Java.</li></ul>
7.2 Obiectivele specifice	<p>Obiectivele specifice ale acestei discipline sunt:</p> <ul style="list-style-type: none"><li>• Însușirea cunoștințelor despre metodologiile de realizare a sistemelor soft;</li><li>• Familiarizarea cu conceptele și preocupările moderne în scrierea softului de aplicație;</li><li>• Însușirea cunoștințelor despre tipurile de modele și instrumentele de modelare folosite în dezvoltarea sistemelor de programare.</li></ul>

## 8. Conținuturi<sup>11</sup>

8.1 Curs	Număr de ore	Metode de predare <sup>12</sup>
1. Etapele realizării unui sistem de programare 1.1. Ce se înțelege prin ingineria sistemelor de programare? 1.2. Ciclul de viață al sistemului de programare, tipuri de abordări în rezolvarea de probleme 1.3. Etapa de analiză 1.4. Etapa de proiectare 1.5. Etapa de implementare 1.6. Etapa de testare	2	Studentii au acces la curs în format electronic. Se vor utiliza atât prezentări interactive cât și tradiționale. Se vor folosi: problematizarea, studiu de caz, conversația.
2. Metodologii de dezvoltare a programelor 2.1. Metodologii generice 2.2. Metodologii concrete	2	
3. Managementul unui sistem de programare 3.1. Funcțiile managementului 3.2. Managementul software 3.3. Managementul configurației 3.4. Managementul echipei	4	
4. Estimarea costului unui sistem de programare 4.1. Modele algoritmice clasice 4.2. Modele algoritmice moderne 4.3. Distribuirea forței de muncă în timp	2	
5. Analiza și proiectarea orientată pe obiecte a sistemelor de programare 5.1. Principiile analizei orientate pe obiecte. 5.2. Metode de analiză orientată obiect. Analiza cerințelor utilizând UML 5.3. Metoda de analiză Coad-Yourdon 5.4. Caracteristicile unei proiectări orientate pe obiecte corecte 5.5. Etapele proiectării orientate pe obiecte. Diagrame UML specifice fazei de proiectare 5.6. Metoda de proiectare Coad-Yourdon	6	
6. Reutilizarea resurselor de programare 6.1. Reutilizarea produselor intermediare 6.2. Reutilizarea instrumentelor 6.3. Programarea generică în Java. Colecții generice de date 6.4. Șabloane de proiectare în Java	8	
7. Verificarea și validarea sistemelor de programare 7.1. Testarea programelor. Criterii de alegere a datelor de test 7.2. Instrumente de testare: JUnit 7.3. Verificarea modelelor. Inspectarea tuturor subproduselor 7.4. Testarea modulelor și a sistemului de programare final 7.5. Verificarea documentației 7.6. Validarea programelor 7.7. Certificarea produselor de programare. Planificarea verificării și validării	4	
Bibliografie <sup>13</sup> 1. Sabin Goron, Elemente de ingineria produselor software, Editura Risoprint, Cluj-Napoca, 2006 2. Dorin Bocu, Inițiere în ingineria sistemelor soft, Editura Albastră, Cluj-Napoca, 2002 3. Militon Frențiu, Verificarea și validarea sistemelor soft, Presa Universitară Clujană, Cluj-Napoca, 2010 4. Philip Wadler, Maurice Naftali, Java Generics and Collections, O'Reilly Media, 2013 5. John Hunt, Guide to the Unified Process Featuring UML, Java and Design Patterns, Springer London Ltd, 2014		

<sup>11</sup> Se detaliază toate activitățile didactice prevăzute prin planul de învățământ (tematicile prelegerilor și ale seminariilor, lista lucrărilor de laborator, conținuturile etapelor de elaborare a proiectelor, tematica fiecărui stagiu de practică). Titlurile lucrărilor de laborator care se efectuează pe standuri vor fi însoțite de notația „(\*)”.

<sup>12</sup> Prezentarea metodelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

<sup>13</sup> Cel puțin un un titlu trebuie să aparțină colectivului disciplinei iar cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, de circulație națională și internațională, existentă în biblioteca UPT.

6. Hamill Paul, Unit Test Frameworks, O'Reilly Media, 2016		
<b>8.2 Activități aplicative<sup>14</sup></b>	<b>Număr de ore</b>	<b>Metode de predare</b>
1. Elaborarea diagramelor specifice fazei de analiză utilizând NetBeans IDE și pluginul easyUML	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
2. Elaborarea diagramelor specifice fazei de proiectare utilizând NetBeans IDE și pluginul easyUML	3	
3. Elaborarea diagramelor specifice fazei de implementare utilizând NetBeans IDE și pluginul easyUML	3	
4. Analiza, proiectarea și implementarea de aplicații în Java ce utilizează clase șablon	3	
5. Analiza, proiectarea și implementarea de aplicații în Java ce utilizează colecții generice	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
6. Analiza, proiectarea și implementarea de aplicații în Java ce utilizează șabloane de proiectare	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
7. Testare folosind biblioteca JUnit	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
Bibliografie <sup>15</sup>		
1. Anca Daniela Ioniță, Alexandra Cernian, Noțiuni aplicative de inginerie a sistemelor de programare, Editura MatrixRom, București, 2009		
2. Sabin Goron, Elemente de inginerie soft, Editura Risoprint, Cluj-Napoca, 2006		
3. Gilad Bracha, Generics in the Java Programming Language, 2004		
4. Vaskaran Sarcar, Java Design Patterns, APRESS, 2016		
5. Beck Kent, JUnit Pocket Guide, O'Reilly Media, 2016		

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

<ul style="list-style-type: none"> <li>Disciplina vine în întâmpinarea așteptărilor angajatorilor reprezentativi din domeniul aferent programului prin conținutul orelor de curs și laborator.</li> </ul>
---

**10. Evaluare**

Tip activitate	10.1 Criterii de evaluare <sup>16</sup>	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	Oral - subiecte teoretice și aplicații	0,66
10.5 Activități aplicative	<b>S:</b>		
	<b>L:</b> Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator	Oral – aplicații utilizând calculatorul	0,34
	<b>P<sup>17</sup>:</b>		
	<b>Pr:</b>		
<b>10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor<sup>18</sup>)</b>			

<sup>14</sup> Tipurile de activități aplicative sunt cele precizate în nota de subsol 5. Dacă disciplina conține mai multe tipuri de activități aplicative atunci ele se trec consecutiv în liniile tabelului de mai jos. Tipul activității se va înscrie într-o linie distinctă sub forma: „Seminar:”, „Laborator:”, „Proiect:” și/sau „Practică:”.

<sup>15</sup> Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

<sup>16</sup> Fișele disciplinelor trebuie să conțină procedura de evaluare a disciplinei cu precizarea criteriilor, a metodelor și a formelor de evaluare, precum și cu precizarea ponderilor atribuite acestora în nota finală. Criteriile de evaluare se formulează în mod distinct pentru fiecare activitate prevăzută în planul de învățământ (curs, seminar, laborator, proiect). Ele se vor referi și la formele de verificare pe parcurs (teme de casă, referate ș.a.)

<sup>17</sup> În cazul când proiectul nu este o disciplină distinctă, în această rubrică se va preciza și modul în care rezultatul evaluării proiectului condiționează admiterea studentului la evaluarea finală din cadrul disciplinei.

<sup>18</sup> Nu se va explica cum se acorda nota de promovare.

- La finalul cursului, respectiv al laboratorului, studenții trebuie să aibă cunoștințe teoretice și abilități de cercetare, strict necesare viitorilor specialiști dovedind competențe în analiza, proiectarea, implementarea și testarea sistemelor informatice dezvoltate în limbajul de programare Java.

**Data completării**

04.09.2017

**Director de departament  
(semnătura)**

.....

**Titular de curs  
(semnătura)**

.....

**Data avizării în Consiliul Facultății<sup>19</sup>**

06.09.2017

**Titular activități aplicative  
(semnătura)**

.....

**Decan  
(semnătura)**

.....

---

<sup>19</sup> Avizarea este precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii cu privire la fișa disciplinei.