

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnică Timișoara
1.2 Facultatea ² / Departamentul ³	Facultatea de Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod ⁴)	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 ⁵	Programare Java / DS						
2.2 Titularul activităților de curs	Prof. dr. ing. Pănoiu Manuela						
2.3 Titularul activităților aplicative ⁶	Șef lucr. dr. ing. Ghiormez Loredana						
2.4 Anul de studii ⁷	III	2.5 Semestrul	I	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei ⁸	DI

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

3.1 Număr de ore asistate integral/săptămână	5 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	- /2/1
3.1* Număr total de ore asistate integral/sem.	63 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	- /28/7
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	4.43 , 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
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			2.43
3.7* Număr total de ore activități neasistate/semestru	62 , 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			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			34
3.8 Total ore/săptămână ¹⁰	9.43				
3.8* Total ore/semestru	125				
3.9 Număr de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Cunoștințe de algoritmică și programarea calculatoarelor
4.2 de competențe	• Operare pe calculator

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

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina.

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

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

⁵ 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).

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

⁷ Anul de studii în care este prevăzută disciplina în planul de învățământ.

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

⁹ 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.

¹⁰ 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	• Sală de curs, Materiale suport: laptop, proiector, tablă.
5.2 de desfășurare a activităților practice	• Laborator cu 12-14 calculatoare, tablă.

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">• C 2. Utilizarea sistemelor informatice de prelucrare și gestiune a datelor.• C 6. Configurarea, implementarea și folosirea sistemelor de achiziție de date.
Competențele transversale în care se înscriu competențele specifice	<ul style="list-style-type: none">•

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

7.1 Obiectivul general al disciplinei	• Dobândirea de abilități de programare în limbajul Java
7.2 Obiectivele specifice	<p>Dobândirea de abilitati:</p> <ul style="list-style-type: none">• de programare într-un limbaj de programare de generația a 4 - a care oferă facilitati de programare vizuala dirijata de evenimente – Limbajul Java – și deprinderea de a lucra cu structurile de control ale acestui limbaj• de a implementa aplicații ce pot lucra în rețea• de a implementa aplicații cu fluxuri de date• de a implementa aplicații cu fire de execuție• de a accesa baze de date din aplicații Java

8. Conținuturi¹¹

8.1 Curs	Număr de ore	Metode de predare ¹²
1. Introducere în Java 1.1. Platforme de lucru Java; 1.2. Structura lexicală a limbajului Java 1.3. Tipuri de date și variabile 1.4. Controlul execuției 1.5. Vectori 1.6. Șiruri de caractere	2	Expunerea informațiilor esențiale pe videoproiector. Detalierea informațiilor expuse Demonstrarea utilizând un mediu de programare Java Conversația
2. Obiecte și clase 2.1. Crearea, folosirea și distrugerea obiectelor 2.2. Tipuri referință 2.3. Tablouri de variabile.	2	
3. Declarația unei noi clase de obiecte 3.1. Variabilele unei clase 3.2. Modificatori 3.3. Protecție 3.4. Variabile predefinite: this și super 3.5. Metodele unei clase. Metode abstracte; Metode finale; 3.6. Protejarea metodelor 3.7. Derivarea claselor	4	
4. Modele de programare 4.1. Aplicații independente; applet-uri 4.2. Structura programelor Java: pachete de clase; importul claselor	2	
5. Excepții 5.1. Ierarhia claselor ce definesc excepții 5.2. "Prinderea" și tratarea excepțiilor 5.3. "Aruncarea" excepțiilor 5.4. Definiții de excepții proprii.	2	
6. Intefețe 6.1. Ce reprezintă interfața . Utilizarea interfețelor 6.2. Moștenire multiplă cu ajutorul interfețelor	2	
7. Organizarea claselor 7.1. Pachete 7.2. Organizarea fișierelor 7.3. Arhive JAR	2	
8. Desenarea 8.1. Suprafețe de desenare 8.2. Primitive grafice 8.3. Folosirea fonturilor 8.4. Folosirea culorilor 8.5. Folosirea imaginilor	2	
9. Fire de execuție 9.1. Crearea unui fir de execuție 9.2. Stările unui fir de execuție 9.3. Gruparea firelor de execuție 9.4. Clasele Timer și TimerTask	2	
10. Interfața grafică cu utilizatorul 10.1. Modelul AWT 10.2. Administrarea componentelor 10.3. Tratarea evenimentelor 10.4. Folosirea ferestrelor 10.5. Folosirea meniurilor	4	
11. Intrări și ieșiri 11.1. Fluxuri 11.2. Utilizarea fluxurilor 11.3. Intrări și ieșiri cu format 11.4. Fluxuri standard de intrare/ieșire	2	
12. Pachetul Swing	2	

¹¹ 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 „(*)”.

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

Bibliografie ¹³		
<ol style="list-style-type: none"> 1. Curs format electronic de pe campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=3577 2. Iordan A. E., Pănoiu M., Programarea Avansata A Intefetelor Grafice Utilizator, Editura Politehnica Timișoara, 2013 3. Tanasa S., Olaru C., Java de la 0 la expert, editura Polirom, Colectia Calculatoare. Informatica, 2011. 5. Cristian Frasinariu, Curs practic de Java, Editura MatrixRom, 2005. 		
8.2 Activități aplicative¹⁴	Număr de ore	Metode de predare
LABORATOR	28	Verificare cunoștințe din tematica laboratorului Elaborare aplicații și testare în NetBeans
1. Mediul de programare NetBeans. Aplicații Java care prezintă conceptul de clasă.	2	
2. Aplicații Java independente care ilustrează conceptul de derivare prin moștenire.	2	
3. Aplicații Java independente care ilustrează conceptul de compunere a obiectelor.	2	
4. Aplicații Java care utilizează componentele de tip Canvas.	2	Verificare cunoștințe din tematica laboratorului Elaborare aplicații și testare în NetBeans
5. Aplicații care utilizează clasa Graphics.	4	
6. Animații Java cu fire de execuție.	4	Verificare cunoștințe din tematica laboratorului Elaborare aplicații și testare în NetBeans.
7. Aplicații care utilizează componente AWT. Tratarea evenimentelor AWT. 8. Aplicații care utilizează pachetul javax.swing. Tratarea evenimentelor.	4	Verificare cunoștințe din tematica laboratorului Elaborare aplicații și testare în NetBeans.
9. Aplicații ce utilizează fluxuri de date. 10. Baze de date în Java.	4 4	Verificare cunoștințe din tematica laboratorului Elaborare aplicații și testare în NetBeans.
PROIECT Studentii vor implementa o aplicație tip proiect în care vor aplica principalele concepte învățate: clase abstracte, moștenire, compunere, interfețe, tratarea erorilor, fișiere, baze de date, interfața grafică și tratarea evenimentelor	7	Verificare săptămânală stadiu implementare proiect
Bibliografie ¹⁵		
<ol style="list-style-type: none"> 1. Ghiormez L., Laborator în format electronic, Campus Virtual UPT, https://cv.upt.ro/course/view.php?id=3577 2. Tanasa S., Olaru C., Java de la 0 la expert, editura Polirom, Colectia Calculatoare. Informatica, 2011. 3. Danciu. D, Mardale G., Arta programarii in Java , Editura Alabastra, 2003. 4. Iordan A., Panoiu M., Programarea avansata a interfetelor grafice utilizator in Java , Editura Politehnica Timisoara, Colectia Calculatoare, 2013. 		

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

- Disciplina este din categoria disciplinelor fundamentale, și se regăsește în planul de învățământ (eventual sub denumiri apropiate) al tuturor programelor de studii din domeniul fundamental Științe Inginerești din România. Este o disciplină solicitată de toate companiile cu profil IT.

Disciplina se regăsește și în planul de învățământ al programelor de studii din străinătate :

- <https://www.bradley.edu/academic/departments/electrical/programs/ee/>

10. Evaluare

¹³ 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.

¹⁴ 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:”.

¹⁵ Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

Tip activitate	10.1 Criterii de evaluare ¹⁶	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Înțelegerea corespunzătoare a principiilor programării pe platforma Java, abilitatea de a elabora soluții tehnice clare pentru probleme specifice și cunoașterea temeinică a elementelor fundamentale ale sintaxei și tehnologiilor Java.	Examenul se desfășoară on-line pe campusul virtual al UPT. Și constă din mai multe subiecte cu caracter aplicativ	66 %
10.5 Activități aplicative	S:		
	L: Lucrări de control, teme de casă și răspunsurile la întrebările puse la laborator	Lucrări de control pe calculator, Teme pe suport electronic	17%
	P ¹⁷ : Criteriile de evaluare vor fi bazate pe elementele implementate în proiect (clase abstracte, moștenire, compunere, fișiere/baze de date)	Prezentare orală a proiectului folosind powerpoint și Netbeans, întrebări din cod sursă	17%
	Pr:		
10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁸)			
Pentru a promova laboratorul trebuie ca studentul să participe la toate testele de laborator, iar media notelor obținute în urma testelor de laborator să fie minim 5.			
<ul style="list-style-type: none"> Pentru a promova examenul studentul trebuie să obțină jumătate din punctajul total alocat evaluării on-line la examen. Pentru a promova disciplina trebuie ca ambele activități (curs și laborator) să fie promovate. 			

Data completării

05.10.2023

**Director de departament
(semnătura)**

.....



**Titular de curs
(semnătura)**

.....


Data avizării în Consiliul Facultății¹⁹

16.10.2023

**Titular activități aplicative
(semnătura)**

.....


**Decan
(semnătura)**

.....


¹⁶ 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.)

¹⁷ 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.

¹⁸ Nu se va explica cum se acorda nota de promovare.

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