

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea ¹ / Departamentul ²	Inginerie Hunedoara/Inginerie Electrică și Informatică Industrială
1.3 Domeniul de studii (denumire/cod ³)	Inginerie electrică / 90
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Inginerie electrică și calculatoare / 60 / Inginer

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă ⁴	Programare Java / DS						
2.1b Denumirea disciplinei în limba engleză	Java Programming						
2.2 Titularul activităților de curs	Prof. dr. Ing. Pănoiu Manuela						
2.3 Titularul activităților aplicative ⁵	Șef lucr. dr. ing. Ghiormez Loredana, As.dr.ing. Raț Cezara Liliana						
2.4 Anul de studii ⁶	III	2.5 Semestrul	I	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei ⁷	DO

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁸

3.1 Număr de ore asistate integral/săptămână	5 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	0/2/1
3.1* Număr total de ore asistate integral/sem.	70 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	0/28/14
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	3.93 , 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			1.93
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			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			27
3.8 Total ore/săptămână ⁹	8.93				
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 rezultatele învățării	• Operare pe calculator

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. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	C3. Studentul/absolventul explică și interpretează desenele care detaliază proiectarea produselor, a instrumentelor și a sistemelor de inginerie electrică
Abilități	- A3. Studentul/absolventul specifică proprietăți tehnice ale bunurilor, materialelor, metodelor, proceselor, serviciilor, sistemelor, software-ului și funcționalităților, prin identificarea și răspunsul la nevoile particulare care urmează să fie satisfăcute în funcție de cerințele clienților - A5. Studentul/absolventul desenează schițe și proiectează sisteme, produse și componente electrice utilizând programe și echipamente informatice de proiectare asistată de calculator (CAD). Desenează schițe ale panourilor electrice, scheme electrice, diagrame de cablare electrică și alte detalii ale ansamblului. Creează schițe și desene tehnice prin utilizarea de software specializat. - A8. Elaborează documentații tehnice, rapoarte și prezentări clare și corect structurate, în limba română și engleză
Responsabilitate și autonomie	- RA2. Studentul/absolventul efectuează căutări bibliografice în literatura de specialitate, consultă și folosește bazele de date științifice și alte surse de informare din domeniul ingineriei electrice. - RA5. Studentul/absolventul reflectă în mod critic, reflexiv, cu simțul responsabilității și în spirit democratic asupra responsabilităților etice și sociale legate de managementul activităților din domeniul ingineriei electrice, de luarea deciziilor și de formularea opiniilor - RA7. Își evaluează constant propriile nevoi de dezvoltare profesională și adoptă o atitudine proactivă față de învățarea pe tot parcursul vieții.

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

- Dobândirea de abilități de programare în limbajul Java

Dobândirea de abilitati:

- de programare într-un limbaj de programare de generația a 4 - a care oferă facilități de programare vizuală dirijată 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. Conținuturi¹⁰

8.1 Curs	Număr de ore	Metode de predare ¹¹
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. 2.4. Clase înfășurătoare	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" ,si tratarea excepțiilor 5.3. "Aruncarea" excepțiilor 5.4. Definiri 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	
Bibliografie ¹² Pănoiu M., Programare Java note de curs , https://www.fih.upt.ro/intranet/user/md/ 1. Iordan A. E., Pănoiu M., Programarea Avansata A Intefetelor Grafice Utilizator, Editura Politehnica Timișoara, 2013 2. Tanasa S., Olaru C., Java de la 0 la expert, editura Polirom, Colectia Calculatoare. Informatica, 2011. 3. Reinhold Haipl, Java pentru ingineri, Editura Mirton, Timișoara 2003 4. Cristian Frasinariu, Curs practic de Java, Editura MatrixRom, 2005		
8.2 Activități aplicative ¹³	Număr de ore	Metode de predare
1 Mediul de programare NetBeans. Aplicații Java care prezintă conceptul de clasă	2	Verificare cunoștințe din tematica laboratorului. Elaborare aplicații și testare în NetBeans
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	
5. Aplicații care utilizează clasa Graphics.	4	Verificare cunoștințe din tematica laboratorului. Elaborare aplicații și testare în NetBeans
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	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, interfețe, tratarea erorilor, fișiere, baze de date, interfața grafică utilizator și tratarea evenimentelor	14	Verificare săptămânală stadiu implementare proiect
Bibliografie ¹⁴ 1.Tanasa S., Olaru C., Java de la 0 la expert, editura Polirom, Colectia Calculatoare. Informatica, 2011. 2. Danciu. D, Mardale G., Arta programarii in Java , Editura Alabastra, 2003. 3.. Iordan A., Panoiu M., Programarea avansata a interfetelor grafice utilizator in Java , Editura Politehnica Timisoara, Colectia Calculatoare, 2013 4. Ghiormez L., Laborator și proiect în format electronic – Programare Java, 2024, Campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=3576		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ¹⁵	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	- Înțelegerea conceptelor fundamentale ale limbajului Java. - Explicarea principiilor POO (încapsulare, moștenire, polimorfism). - Utilizarea corectă a claselor, metodelor și tratarea excepțiilor. - Capacitatea de a analiza cerințele unei probleme și de a propune soluții		
9.5 Activități aplicative	S:		
	L: - Implementarea corectă a aplicațiilor Java conform cerințelor. - Aplicarea principiilor OOP în realizarea proiectelor. - Utilizarea colecțiilor, fișierelor și excepțiilor în programe funcționale. - Prezentarea și argumentarea soluțiilor proprii în cadrul activităților practice.		
	P ¹⁶ :		
	Pr:		
9.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁷)			
Studentul trebuie să demonstreze că poate scrie, compila și executa programe simple în Java, aplicând corect principiile OOP, structurile de control, colecțiile și tratarea excepțiilor. Promovarea se obține prin realizarea unei aplicații funcționale și stăpânirea noțiunilor de bază teoretice și practice.			

Data completării

Titular de curs
(semnătura)

Titular activități aplicative
(semnătura)

10.09.2025



**Director de departament
(semnătura)**



Data avizării în Consiliul Facultății¹⁸

17.09.2025

**Decan
(semnătura)**

