# FIȘA DISCIPLINEI

## 1. Date despre program

| | |
|---|---|
| 1.1 Instituția de învățământ superior | **Universitatea Națională de Știință și Tehnologie POLITEHNICA București** |
| 1.2 Facultatea | **Ştiinţe Aplicate** |
| 1.3 Departamentul | Matematică-Informatică |
| 1.4 Domeniul de studii universitare | Științe Inginerești Aplicate |
| 1.5 Programul de studii universitare | Matematică şi Informatică Aplicată în Inginerie |
| 1.6 Ciclul de studii universitare | Licență |
| 1.7 Limba de predare | Română |
| 1.8 Locația geografică de desfășurare a studiilor | București |

## 2. Date despre disciplină

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1 Denumirea disciplinei | | | | | | | |
| (ro) | | | | **Programare orientată pe obiecte** | | | |
| (en) | | | | **Object oriented programming** | | | |
| 2.2 Titularul activităților de curs | | | | **Conf. Univ. Dr. MARICA Aurora-Mihaela** | | | |
| 2.3 Titularul activităților de laborator/seminar | | | | **Conf. Univ. Dr. MARICA Aurora-Mihaela** | | | |
| 2.4 Anul de studiu | 2 | 2.5 Semestrul | I | 2.6. Tipul de evaluare | V | 2.7 Statutul disciplinei | Ob |
| 2.8 Categoria formativă | | DS | 2.9 Codul disciplinei | | UPB.13.S.03.O.126 | | |

## 3. Timpul total (ore pe semestru al activităților didactice)

| | | | | | |
|---|---|---|---|---|---|
| 3.1 Număr de ore pe săptămână | 4 | Din care: 3.2 curs | 2 | 3.3 seminar/laborator | 1/1 |
| 3.4 Total ore din planul de învățământ | 56 | Din care: 3.5 curs | 28 | 3.6 seminar/laborator | 28 |
| **Distribuția fondului de timp:** | | | | | |
| Studiul după manual, suport de curs, bibliografie și notițe | | | | | 8 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate | | | | | 8 |
| Pregătire teme și referat | | | | | 47 |
| Tutorat | | | | | 3 |
| Examinări | | | | | 3 |
| 3.7 Total ore studiu individual | **69** | | | | |
| 3.8 Total ore pe semestru | **125** | | | | |
| 3.9 Numărul de credite | **5** | | | | |

## 4. Precondiții (acolo unde este cazul)

| | |
|---|---|
| 4.1 de curriculum | Parcurgerea și promovarea disciplinei Programarea calculatoarelor și limbaje de programare 1/2, în care se însușesc noțiuni și se formează deprinderi de utilizare a programului C++ |
| 4.2 de rezultate ale învățării | Acumularea unor noțiuni generale de programare cum ar fi instrucțiune sau structuri de control (if, for, while) |

## 5. Condiții necesare pentru desfășurarea optimă a activităților didactice

| | |
|---|---|
| 5.1 Curs | Cursul se va desfășura într-o sală dotată cu videoproiector și conexiune la internet pentru calculatorul/laptopul cadrului didactic |
| 5.2 Seminar/Laborator | Laboratoarele/seminariile se vor desfășura într-o sală cu dotare specifică, care trebuie să includă programele JDK17 și IntelliJ instalate pe calculatoarele studenților și conexiune la internet |

**6. Obiectiv general.** Această disciplină se studiază în cadrul specializării Matematică și Informatică Aplicată în Inginerie și își propune să familiarizeze studenții cu principalele noțiuni de programare orientată pe obiecte și să le formeze abilitățile de folosire a limbajului Java în mediul de programare IntelliJ. Disciplina abordează ca tematică specifică următoarele noțiuni fundamentale: clase și obiecte, extinderea claselor, polimorfism, tablouri, clase abstracte, interfețe, excepții, fluxuri de intrare/ieșire, colecții de date, expresii lambda, fire de execuție, socketuri, interfețe grafice, lucrul cu baze de date, servleturi, restful web services, toate acestea contribuind la formarea de către studenți a unei viziuni de ansamblu asupra reperelor metodologice/procedurale aferente domeniului.

**7. Rezultatele învățării**

| | |
|---|---|
| **Cunoștințe** | C1. Identifică și descrie concepte, principii și metode de bază din informatică.<br>C2. Explică și interpretează rezultate teoretice și experimentale din informatică.<br>C8. Recunoaște, explică și specifică fundamentele matematice aplicate în informatică.<br>C9. Alege, descrie, analizează, explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă.<br>• C1: Să diferențieze noțiunile de source/byte code, fișierele de tip .java și .class<br>• C1: Să explice funcțiile pentru compiler, JVM - Java virtual machine, interpreter, JRE – Java Runtime Environment, JDK – Java Development Kit, mediu de dezvoltare IDE – Integrated Development Environment<br>• C1: Să exemplifice noțiunea de literal întreg, în virgulă mobilă, boolean, caracter, stringuri - șir de caractere, null, keywords<br>• C1: Să exemplifice membri ai fiecărui tip de date primitive float/double, byte/short/int/long, char, boolean și clasele wrapper/înfășurătoare corespunzătoare<br>• C1: Să folosească corect operatorii în instrucțiuni<br>• C1: Să folosească corect instrucțiunile de tip decision making if/else if/else sau switch, looping for și while, branching break/continue/return<br>• C1: Să explice utilitatea unui pachet de clase și modul de importare a lui într-un program<br>• C1: Să explice/exemplifice noțiunea de referință, tablou uni/bidimensional<br>• C1: Să enunțe/exemplifice sintaxa unei clase/metode/interfețe (interfețe marker)<br>• C1: Să enunțe/exemplifice sintaxa unei clase abstracte/imutabile/record/adaptor/sealed<br>• C1: Să explice funcționalitatea modificatorilor de acces public, protected, private<br>• C1: Să explice funcționalitatea modificatorilor static, final<br>• C1: Să explice/exemplifice noțiunile de abstractizare, încapsulare, moștenire, polimorfism, overloading, overriding, constructor, agregare, compoziție și utilizarea cuvintelor-cheie new, this, extends, abstract<br>• C1: Să diferențieze clasele de stringuri String, StringBuider și StringBuffer și metodele lor<br>• C1: Să recunoască principalele tipuri de excepții din Java și să le prindă/trateze corespunzător cu instrucțiuni de tip try-catch sau throws<br>• C2: Să creeze/manipuleze colecții de date (enum, list, set, queue, map, iterator)<br>• C2: Să deschidă fluxuri/stream-uri de intrare/ieșire<br>• C2: Creează socket-uri simple pentru a comunica între mai multe calculatoare dintr-o rețea<br>• C2: Să creeze fire de execuție simple și să le pornească în diferite contexte<br>• C2: Să utilizeze baze de date pentru a gestiona datele unei aplicații<br>• C2: Să definească expresii lambda și să le utilizeze în cadrul programelor<br>• C8: Să utilizeze corect metodele clasei Math. |

| | |
|---|---|
| **Abilități** | • A1. Operează cu concepte, principii și metode de bază din informatică.<br>• A9. Proiectează aplicații funcționale de complexitate mică/medie în limbajul Java.<br>• A18. Propune și implementează metode de dezvoltare a proiectelor informatice complexe. |
| **Responsabilitate și autonomie** | • R1. Aplică valorile eticii și deontologiei profesiei de inginer: Respectă principiile de etică academică, citând corect sursele bibliografice utilizate.<br>• R4. Practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor.<br>• R5. Comunică eficient despre activitățile de programare cu o gamă largă de public.<br>• R6. Este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. Identifică oportunități de formare continuă și de utilizare eficientă, pentru propria dezvoltare, a surselor informaționale (portaluri Internet, aplicații software de specialitate, baze de date, cursuri on-line etc.), atât în limba română, cât și într-o limbă de circulație internațională.<br>• R7. Selectează/analizează surse bibliografice potrivite proiectelor pe care le realizează.<br>• R8. Produce software și îl adaptează continuu la noile tehnologii și cerințe de piață. Analizează/valorifică oportunități de afaceri în domeniul programării.<br>• R9. Demonstrează autonomie în învățare și în organizarea situației/contextului de învățare sau a situației-problemă de rezolvat.<br>• R11. Își asumă în mod responsabil sarcinile profesionale și respectă normele de etică și deontologie profesională: Demonstrează receptivitate pentru contexte noi de învățare.<br>• R12. Lucrează eficient ca membru în echipă sau lider al acesteia. Manifestă colaborare cu ceilalți colegi/profesori în desfășurarea activităților didactice. Demonstrează abilități de management al situațiilor din viața reală (gestionarea timpului, colaborare/conflict) |

## 8. Metode de predare

Pornindu-se de analiza caracteristicilor de învățare ale studenților și de la nevoile lor specifice, procesul de predare va explora metode de predare atât expozitive (prelegerea, expunerea, prezentarea), cât și conversative-interactive (conversația euristică, problematizarea, modelarea), bazate pe modele de învățare prin descoperire facilitate de explorarea directa și indirectă a realității (exemplificarea, demonstrația), dar și pe metode bazate pe acțiune, precum activitățile practice.

În activitatea de predare vor fi utilizate prelegeri la tablă, asociate când este cazul cu prezentări Power Point sau diferite filmulețe care vor fi puse la dispoziția studenților. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noțiunilor parcurse la ultimul curs. Prezentările utilizează și imagini/scheme, astfel încât informațiile prezentate să fie mai ușor de înțeles/asimilat.

Această disciplină acoperă informații și activități practice menite să-i sprijine pe studenți în dezvoltarea unor relații de colaborare și comunicare într-un climat favorabil învățării prin descoperire.

Se va avea în vedere exersarea abilităților de ascultare activă şi de comunicare asertivă, precum şi a mecanismelor de construcție a feedback-ului, ca modalități de reglare comportamentală în situații diverse și de adaptare a demersului pedagogic la nevoile de învățare ale studenților.

Se va exersa abilitatea de lucru în echipă pentru rezolvarea diferitelor sarcini de învățare.

## 9. Conținuturi

| **CURS** | | |
|---|---|---|
| CAPITOL | CONȚINUT | #ORE |
| I. | Prezentarea generală a platformei Java, structura unui program simplu, afișarea/citirea datelor în/din terminal | 2 |
| II. | Pachete de clase, tipuri de date/operatori, literali, instrucțiuni | 2 |

| | | |
|---|---|---|
| III. | Referințe, tablouri, clase/obiecte/metode, modificatori de acces/clasă, date membre/câmpuri, constructori, singleton pattern | 2 |
| IV. | Extinderea claselor, moștenire | 2 |
| V. | Agregare și compoziție, clase de String-uri, clase imutabile/records | 2 |
| VI. | Interfețe | 2 |
| VII. | Interfețe marker, clase sealed | 2 |
| VIII. | Enumerări și gestionarea excepțiilor | 2 |
| IX. | Colecții | 2 |
| X. | Fluxuri de intrare/ieșire | 2 |
| XI. | Serializarea/externalizarea obiectelor, socket-uri, fire de executare | 2 |
| XII. | JDBC - Java DataBase Connectivity | 2 |
| XIII. | Programare funcțională/expresii Lambda | 2 |
| XIV. | Stream-uri | 2 |

Bibliografie

1. A. Marica, Programare orientată pe obiecte, suport electronic de curs
2. J. Bloch, Effective Java (3rd edition), Addison-Wesley Professional, 2018
3. R.G. Urma, M. Fusco, A. Mycraft, Modern Java in action, Manning, 2018
4. R.G. Urma, M. Fusco, A. Mycraft, Java 8 in action, Manning, 2014
5. B. Eckel, Thinking in Java, PrenticeHall, 2012
6. B. Eckel, On Java 8, Leanpub, 2021, ISBN 978-0-9818725-2-0
7. Ș. Tanasă, C. Olaru, Ș. Andrei, Java de la 0 la expert, Ed. Polirom, 2011
8. C. Frăsinaru, Curs practic de Java, Matrix Rom Bucuresti, 2005
9. C.S. Horstmann, Core Java, vol. I – Fundamentals, Addison-Wesley, Pearson Education, 2019
10. R. Cadenhead, SamsTeach Yourself Java in 21 Days, Pearson Education, 2020
11. **H. Schildt, Java – The Complete Reference, XII Edition, McGraw Hill, 2022**
12. Tutoriale Java pe site-ul Oracle
13. Tutoriale Java Tutorials Point
14. Tutoriale Java canal Youtube Math and Science, vol1, vol2, vol3, vol4
15. Tutoriale Java canal Youtube Facem Soft
16. Java tutorial for beginners, Programming with Mosh
17. Java programming full course, Simplilearn
18. Laboratoare POO – Facultatea de Automatică, UPB, ocw.cs.pub.ro/courses/poo-ca-cd
19. Laboratoare Tehnologii Java – Universitatea din Craiova
20. Programare orientată pe obiecte – laboratoare Universitatea Tehnică din Cluj-Napoca
21. H.M. Deitel, P.J. Deitel, Java How to Program, 7-th Edition, Prentice Hall, 2007
22. **T. Buchalka, Java 17 Masterclass: Start coding in 2024, curs Udemy**

**SEMINAR**

| CAPITOL | CONȚINUT | #ORE |
|---|---|---|
| I. | Graphics2D, paintComponent | 2 |
| II. | Graphics2D: translații/rotații | 2 |
| III. | JFrame form - calculator științific | 2 |
| IV. | JFrame form - pian virtual | 2 |
| V. | Java Android - calendar | 2 |
| VI. | Construirea unei aplicații web cu Spring Boot | 2 |
| VII. | Construirea unui proiect Maven | 2 |

**LABORATOR**

| | | |
|---|---|---|
| VIII. | Elemente de bază ale limbajului Java - Java basics | 2 |
| IX. | Constructori și referințe | 2 |
| X. | Agregare și moștenire | 2 |

| | | |
|---|---|---|
| XI. | Cuvintele-cheie static și final | 2 |
| XII. | Clase interne | 2 |
| XIII. | Polimorfism, overriding | 2 |
| XIV. | Test laborator | 2 |

Bibliografie

1. A. Marica, Programare orientată pe obiecte, suport electronic de curs
2. Tutoriale Java pe site-ul Oracle
3. Tutoriale Java Tutorials Point
4. Tutoriale Java canal Youtube Math and Science, vol1, vol2, vol3, vol4
5. **Java tutorial for beginners, Programming with Mosh**
6. **Laboratoare POO – Facultatea de Automatică, UPB, ocw.cs.pub.ro/courses/poo-ca-cd**
7. **I.A. Giosan - Programare orientată pe obiecte - curs Universitatea Tehnică din Cluj-Napoca**
8. L. Ammeraal, K. Zhang, Computer Graphics for Java Programmers,3rd Ed, Springer2017

**23. Evaluare.** Activitatea fiecărui student va fi evaluată cu maxim 100p, din care:

- 10p prezența curs+seminar+laborator
- Parcurgerea proiectelor de seminar și efectuarea temelor 30p
- Parcurgerea proiectelor de laborator și efectuarea temelor 30p
- Testare în timpul ultimului laborator 20p
- Răspunsuri/activitate curs 10p

| **Tip activitate** | **Criterii de evaluare** | **Metode de evaluare** | **Pondere în nota finală** |
|---|---|---|---|
| 10.1.Curs | SC=realizare sarcini curs | Conversația euristică | **Total curs=15** (SC=10, prezența=5) |
| 10.2L.Laborator | SL=realizare sarcini laborator | Conversația euristică<br>Depozitare Moodle | **Total laborator=52.5 (SL=30, prezența=2.5, testare=20)** |
| 10.2S. Seminar | SS=realizare sarcini seminar | Conversația euristică<br>Depozitare Moodle | **Total seminar=32.5** (SS=30, prezența=2.5) |
| 10.3 Standard minim de performanţă | | | |
| • Pentru promovarea disciplinei, studentul trebuie să obțină cel puțin 50 de puncte din 100.<br>• Participarea la toate seminariile/laboratoarele (recuperarea zilelor de absență). Participarea la testarea din cadrul ultimului laborator. | | | |

Data completării: 22.09.2025

Semnătura titularului de curs/seminar/laborator:
Conf. Univ. dr. MARICA Aurora-Mihaela

Data aprobării în Consiliul Facultății

Semnătura decanului Facultății de Științe Aplicate:
Conf.dr. PETRESCU-NIȚĂ Alina