

FIȘA DISCIPLINEI
Metode Avansate de Programare

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Metode Avansate de Programare				Codul disciplinei		MLE5008
2.2. Titularul activităților de curs			Conf.dr.ing. Florin Craciun					
2.3. Titularul activităților de seminar			Conf.dr.ing. Florin Craciun					
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligativu	

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

3.1. Număr de ore pe săptămână	6	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2 sem. + 2 lab
3.4. Total ore din planul de învățământ	84	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	28 sem + 28 lab
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					26
Tutoriat (consiliere profesională)					5
Examinări					5
Alte activități					0
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				66	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Programare orientată pe obiecte, Algoritmi si Structuri de date
4.2. de competențe	Noțiuni de bază și abilități de programare

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	proiector
5.2. de desfășurare a seminarului/ laboratorului	proiector

6.1. Competențele specifice acumulate¹

Competențe profesionale/esențiale	• programarea în limbaje de nivel înalt • dezvoltarea și întreținerea aplicațiilor informatice
Competențe transversale	• aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • desfășurarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea capacităților empatice de comunicare interpersonală, de relaționare și colaborare cu grupuri diverse

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: • Absolventul are cunoștințe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor. • Absolventul are abilitatea de a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu. • Absolventul are aptitudinile necesare pentru conceperea programelor de calculator și analiza sistemelor software. • Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții. • Absolventul are abilitatea de a aplica reguli generale unor probleme specifice și de a produce soluții relevante. • Absolventul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații. • Absolventul are cunoștințe legate de programare, matematică, inginerie și tehnologie și are abilitățile necesare pentru a le folosi în crearea de sisteme informatice complexe.
Aptitudini	Studentul este capabil să ... • Absolventul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații. • Absolventul este capabil să aplice șabloane arhitecturale, șabloane de proiectare și bunele practici în domeniu pentru a proiecta aplicații software de complexitate mare.
Responsabilități și autonomie	Studentul are capacitatea de a lucra independent pentru • Absolventul are capacitatea de a evalua diferite arhitecturi și soluții posibile pentru o problemă și a alege pe cel potrivit pentru cerințele și constrângerile specifice aplicației de dezvoltat. • Absolventul are capacitatea de a alege și folosi paradigme de programare (procedural, orientat obiect, funcțional) pentru realizarea de aplicații software adecvate specificului domeniului aplicației dezvoltate.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

7.1 Obiectivul general al disciplinei	Fiecare student trebuie să demonstreze că a dobândit un nivel acceptabil de cunoștințe și înțelegere a subiectului, că este capabil să prezinte aceste cunoștințe într-o formă coerentă, că are obiceiuri corecte de analiză, proiectare și implementare bazate pe modele de proiectare și paradigme generale orientate spre obiect.
7.2 Obiectivele specifice	Studenții trebuie să aibă capacitatea de a utiliza limbajul Java, modele de proiectare și de a crea GUI pentru aplicațiile lor. De asemenea, trebuie să fie capabili să utilizeze concepte orientate pe obiecte în analiza și proiectarea programelor.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Introducere în platforma Java: platformă, sintaxa limbajului, tipuri de date primitive, matrice, clase, interfețe, pachete, enumerări, suprascriere, supraîncărcare, excepții	Prezentare, descriere, explicație, dezbateri și dialog, discutarea studiilor de caz	
Colecții și tipuri generice: clase anonime, polimorfism, conversie	Prezentare, descriere, explicație, dezbateri și dialog, discutarea studiilor de caz	
IO, NIO: fluxuri binare și orientate pe caractere, fișiere, canale și tampoane	Prezentare, descriere, explicație, dezbateri și dialog, discutarea studiilor de caz	
Programare funcțională: expresii lambda, fluxuri	Prezentare, descriere, explicație, dezbateri și dialog, discutarea studiilor de caz	
GUI: componente Java FX, gestionarea evenimentelor	Expunere, descriere, explicație, dezbateri și dialog, discutarea studiilor de caz	
Concurență: thread-uri, executori, futures, gestionarea excepțiilor		
Concurență: metode sincronizate vs. asincrone, metode de apel invers, anulare		
GUI (continuare): FXML, CSS. Metaprogramare: reflecție, serializare		
Introducere în C# și .Net		
Colecții în C#		
Operații IO în C#		
GUI în C#		
Introducere în Scala		
Introducere în Rust		

Bibliografie

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. Specificația limbajului Java™ Java SE 7 Edition.
2. Eckel, B., Thinking in Java, ediția a 4-a, Prentice Hall, 2006
3. Eckel, B.: Gândirea în modele cu Java, 2004. MindView, Inc
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994
5. ***, Tutorial Java, 2013. <http://download.oracle.com/javase/tutorial/>
6. Joseph Albahari și Ben Albahari, C# 4.0 în câteva cuvinte, ediția a patra, O’Reilly, 2022
7. ***, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>

8.2 Seminar / laborator	Metode de predare	Observații
Proiect de bază Java	Conversație, dezbateri, studii de caz, exemple	
Proiect Java: Colecții, Generice		
Proiect Java: IO		
Evaluarea proiectului		
Proiect Java: Programare funcțională		
Proiect Java: gestionarea memoriei		
Evaluarea proiectului		
Proiect Java: GUI		
Evaluarea proiectului		
Proiect Java: Concurență		
Proiect Java: concurență		
Evaluarea proiectului		
Proiect Java: sistem de tipuri		
Evaluarea proiectului		
Evaluarea proiectului		

Bibliografie

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. Specificația limbajului Java™ Java SE 7 Edition.
 2. Eckel, B., Thinking in Java, ediția a 4-a, Prentice Hall, 2006
 3. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Modele de proiectare – Elemente de software orientat pe obiecte reutilizabile, Ed. Addison Wesley, 1994
 4. Joseph Albahari și Ben Albahari, C# 4.0 în câteva cuvinte, ediția a patra, O’Reilly, 2010
 5. ***, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>
- ***, Tutorial Java, 2013. <http://download.oracle.com/javase/tutorial/>

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

- Cursul respectă recomandările IEEE și ACM Curricula pentru studiile de informatică;
- Conținutul cursului este considerat de companiile de software ca fiind important pentru competențele medii de dezvoltare software

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- cunoașterea principiilor de bază ale domeniului;	Examen final scris	25
	aplicarea conceptelor cursului în rezolvarea problemelor	Examen final practic	35
10.5 Seminar/laborator	- înțelegerea conceptelor	Teste seminar	5
	- punerea în practică a conceptelor	Proiect de laborator	35

10.6 Standard minim de performanță

- Cel puțin nota 5 (pe o scară de la 1 la 10) la examenul final scris și la examenul final practic. Cel puțin nota 5 pentru nota finală.
- Reguli:
- **Puteți schimba subgrupul pentru laborator o singură dată, la primul laborator. Trebuie să anunțați profesorul de laborator în legătură cu acest lucru. După primul laborator, nu mai puteți schimba ora de laborator a subgrupului.**
- **Trebuie să prezinți fiecare temă de laborator la termenul limită.**
- **- pentru fiecare sarcină de laborator veți primi o notă între 1 și 10.**
- **- termenul limită pentru fiecare temă de laborator este specificat clar în fișierul text al temei**
- **- dacă întârziati cu o săptămână predarea unei lucrări, puteți obține maximum 7 puncte pentru acea lucrare**
- **- dacă întârziati o sarcină cu mai mult de o săptămână, veți primi automat nota 0 pentru aceasta și nu o mai puteți trimite**
- **- nota finală pentru activitatea de laborator este media aritmetică a notelor la temele de laborator**
- **trebuie să realizați toate temele, deoarece problemele din examenul practic final sunt extensii ale temelor de laborator**
- - sarcinile de laborator constau în principal într-un proiect de amploare care vizează implementarea unui interpret (mașină virtuală) al unui limbaj de programare imperativ concurent
- - la fiecare sarcină de laborator (aproape în fiecare săptămână) veți adăuga regulile și structurile de date necesare pentru a executa una sau mai multe instrucțiuni noi ale limbajului de jucărie
- - interpretul limbajului de jucărie va fi implementat în Java
-
- - un program al sarcinilor de laborator (actualizat periodic) poate fi găsit la LabAssignmentsSchedule.pdf
- **Prima condiție pentru a putea susține examenul final este participarea la minimum 90% din laboratoare și minimum 70% din seminarii. Asta înseamnă că trebuie să participați la minimum 10 seminarii și minimum 12 laboratoare. Vă rugăm să citiți**

următorul document:

- <http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf>
- - Zilele de sărbătoare și 2 octombrie sunt considerate în mod implicit ca fiind participări.
- A doua condiție pentru a putea susține examenul final este obținerea unei note minime de 5 la activitatea de laborator „”.
- Reguli pentru studenții din anii anteriori („Restantieri”): studenții trebuie să participe la laboratoare și seminarii, trebuie să facă temele de laborator și trebuie să promoveze examenul final
- - pentru a promova examenul final, trebuie să aveți:
 - -- cel puțin 5 la examenul teoretic final și
 - -- cel puțin 5 la examenul practic final și
 - -- nota finală trebuie să fie de cel puțin 5
- - puteți promova fie examenul teoretic final, fie examenul practic final, fie niciunul
- Reguli pentru al doilea examen („restanta”): Prima condiție pentru a putea susține examenul final este participarea la minimum 90% din laboratoare și minimum 70% din seminarii. Asta înseamnă că trebuie să participați la minimum 10 seminarii și minimum 12 laboratoare. Vă rugăm să citiți următorul document:

<http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf> Sărbătorile

legale și ziua de 2 octombrie sunt considerate în mod implicit ca fiind participate

A doua condiție pentru a putea susține examenul final este obținerea unei note minime de 5 la activitatea de laborator.

Pentru a promova examenul final al celui de-al doilea semestru, trebuie să aveți:

- cel puțin 5 la examenul teoretic final și
- cel puțin 5 la examenul practic final și
- nota finală este 5

- - puteți promova fie ambele examene finale, teoretic și practic, fie niciunul

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

...

Semnătura titularului de curs

Conf.dr. ing. Florin Craciun

Semnătura titularului de seminar

Conf. dr. ing. Florin Craciun

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".