

FIȘA DISCIPLINEI

Metode avansate de programarea aplicațiilor software

Anul universitar 2025-2026

1. Date despre program

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

2. Date despre disciplină

2.1. Denumirea disciplinei		Metode avansate de programarea aplicatiilor software				Codul disciplinei	MLE5235
2.2. Titularul activităților de curs		Conf. dr. Bocicor Maria Iuliana					
2.3. Titularul activităților de seminar		Conf. dr. Bocicor Maria Iuliana					
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	C	2.7. Regimul disciplinei	Obligatorie

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

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	1 sem 1 lab
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	14+14
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					28
Tutoriat (consiliere profesională)					4
Examinări					7
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				69	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Algoritmi și Programare, Bazele programării orientate obiect, Structuri de date
4.2. de competențe	Cunoștințe medii de programare într-un limbaj de programare de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de clasă cu proiector
5.2. de desfășurare a seminarului/ laboratorului	- Laborator cu calculatoare; Java, C# și limbaje de programare, IntelliJ IDEA/Eclipse, Visual Studio IDE - Sală de clasă cu proiector

6. Competențele specifice acumulate¹

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.

Competențe profesionale / esențiale	- Elaborarea și analiza unor algoritmi pentru rezolvarea problemelor. - Programarea în limbaje de nivel înalt (Java, C#).
Competențe transversale	- Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unor atitudini responsabile față de domeniul didactico-științific, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională. - Utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată, atât în limba română, cât și într-o limbă de circulație internațională.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Să deprindă studentul cu un design orientat pe obiecte pentru probleme de complexitate mică/medie, învățarea limbajului de programare Java și crearea de interfețe grafice.
7.2 Obiectivele specifice	- Utilizarea conceptelor orientate pe obiecte în analiza și proiectarea programelor. - Utilizarea și implementarea soluțiilor în limbajul de programare Java. - Crearea interfețelor grafice pentru cerințele date. - Aplicarea modelelor de proiectare în diverse contexte. - Utilizarea unor clase scrise de alți programatori atunci când își construiesc propriile sisteme.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introduction in Java - Platform - Language syntax - Data types. Arrays - Examples		
2. Classes, inheritance - Classes - Object construction - Methods - Inheritance, polymorphism - Abstract classes, interfaces		
3. Generic types, collections in Java - Generic methods - Type erasure - Generic classes and subtyping - Wildcards - Java Collections Framework		
4. Exceptions, Java I/O, JUnit - Exceptions - Java I/O, streams, serialization		

JUnit		
5. JDBC, Functional programming - JDBC API - Java 8 features: Lambda expressions, Java 8 Streams		
6. Graphical User Interfaces - JavaFX applications, scenes, layouts, UI controls - Events		
7. Graphical User Interfaces - Processing events - Model-View-Controller - FXML		
8. Java Reflection, Concurrency - Java Reflection API - Concurrency: processes, threads, multithreaded programming in Java		
9. Concurrency - Threads in Java - Thread synchronization - Concurrent applications in Java		
10. Design Patterns - Creational patterns - Structural patterns - Behavioural patterns		
11. Design Patterns (cont.), Introduction in C# and .NET		
12. C# and .NET - The .NET Architecture - The C# programming language - Classes in C# - Generics - Delegates - Events - Lambda expressions - LINQ		
13. Revision - Revision of the most important topics covered by the course - Examination guide		

Bibliografie

James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley.

2. Eckel, B. Thinking in Java, 4th edition, Prentice Hall, 2006.

3. Eckel, B. Thinking in Patterns with Java, 2004. MindView, Inc.

4. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Longman Publishing, 1995.

5. The Java Tutorials: <https://docs.oracle.com/javase/tutorial/>

6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010.

8.2 Seminar / laborator	Metode de predare	Observații
1. Simple problems in Java. Classes. Layered architecture.	- Interactive exposure - Explanation - Conversation - Examples - Didactical demonstration	Seminarul este structurat ca o activitate de 2 ore, la fiecare 2 săptămâni.
2. Inheritance, interfaces, packages, iterators.		
3. Generics, collections, exceptions.		
4. Serialization, files, JDBC.		
5. Graphical User Interfaces with JavaFX.		
6. Concurrency, threads.		

7. Design patterns.		
8.3 Laborator		
1. Setting up JDK, JRE and JVM, as well as an IDE of choice. Simple problems in Java.	• Explanation • Conversation	Laboratorul este structurat ca o activitate de 2 ore, la fiecare 2 săptămâni.
2. Layered architecture, generics, exceptions.		
3. Files, serialization, JUnit.		
4. JDBC, functional programming (Java 8 streams).		
5. Laboratory test.		
6. Graphical User Interfaces.		
7. Practical examination.		

Bibliografie

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley.
2. Eckel, B. Thinking in Java, 4th edition, Prentice Hall, 2006.
3. Eckel, B. Thinking in Patterns with Java, 2004. MindView, Inc.
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Longman Publishing, 1995.
5. The Java Tutorials: <https://docs.oracle.com/javase/tutorial/>
6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010.

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

- Conținutul disciplinei este în concordanță cu recomandările ACM pentru domeniul informatică.
- Conținutul disciplinei este considerat de companiile software ca important pentru abilități 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	Corectitudinea și completitudinea cunoștințelor acumulate și capacitatea de a proiecta și implementa programe Java.	Examen scris (C)	30%
10.5 Seminar/laborator	Capabilitatea de a utiliza conceptele discutate pentru a rezolva probleme reale.	Evaluare practică (C)	30%
	Corectitudinea temelor de laborator și teste de laborator.	Programe, documentații. Observarea continuă pe parcursul semestrului. Teste de laborator.	40%

10.6 Standard minim de performanță

- Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaștere și înțelegere a domeniului, că este capabil să exprime cunoștințele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni și de a utiliza cunoștințele în rezolvarea unor probleme în limbajul de programare Java.
- Pentru promovare, este obligatorie prezența la minim 5 seminarii și 6 laboratoare .
- Pentru promovare este ca necesar notele pentru evaluari (examen scris, evaluare practica) sa fie minim 5 și nota finală să fie minim 5.

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

Nu se aplică.

Data completării:
15.04.2025

Semnătura titularului de curs
Conf. Dr. Bocicor Maria Iuliana

Semnătura titularului de seminar
Conf. Dr. Bocicor Maria Iuliana

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.*".