

FIȘA DISCIPLINEI

Introducere în metode avansate de programare

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 Matematică și Informatică al Liniei Maghiare
1.4. Domeniul de studii	Matematică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Matematică informatică (în limba maghiară)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Introducere în metode avansate de programare				Codul disciplinei	MLM5252
2.2. Titularul activităților de curs			conf. univ. dr. Darvay Zsolt				
2.3. Titularul activităților de seminar			conf. univ. dr. Darvay Zsolt				
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	C	2.7. Regimul disciplinei	obligatoriu - DS

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	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					Ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					4
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					28
Tutoriat (consiliere profesională)					3
Examinări					6
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	Nu sunt.
4.2. de competențe	Cunoașterea noțiunilor de bază ale programării orientate pe obiecte. Clase și obiecte în limbajul C++.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs dotată cu tablă și videoproiector.
5.2. de desfășurare a seminarului/ laboratorului	Sală de calculatoare, C++, Java.

6. Competențele specifice acumulate

Competențe profesionale/esențiale	- C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic. - C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază. - C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date. - C1.4 Testarea unor aplicații pe baza unor planuri de test. - C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente.
Competențe transversale	- CT1 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. - CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Aprofundarea programării și proiectării orientată pe obiecte prin dobândirea cunoștințelor privind bazele programării în Java și ale celor mai importante șabloane de proiectare.
7.2 Obiectivele specifice	- Învățarea bazelor programării în Java. - Dobândirea unor metode de proiectare care au fost utilizate cu succes în proiectarea bibliotecilor Java. - Îmbunătățirea cunoștințelor privind analiza și proiectarea sistemelor software. - Studierea unor șabloane de proiectare în vederea structurării rezonabile ale aplicațiilor. - Utilizarea conceptelor orientate pe obiecte ale limbajului de programare Java pentru a crea aplicații care au o interfață grafică și sunt bazate pe evenimente.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Limbajul și platforma Java - caracteristici generale - mașina virtuală Java - pachete în Java - comparația cu C++	Prelegerea, dialogul, problematizarea, explicația	
2. Elemente de bază ale limbajului Java - tipuri primitive și referință, tablouri - conversii - expresii și instrucțiuni - transmiterea parametrilor	Prelegerea, dialogul, problematizarea, explicația	
3. Clase și obiecte - notații UML referitoare la clase și relații dintre ele - șabloane - clase interioare - blocuri de inițializare - eliberarea automată a memoriei	Prelegerea, dialogul, problematizarea, explicația	
4. Clase derivate - declararea claselor derivate, moștenirea - supraîncărcarea și ascunderea metodelor - întoarcere covarinată	Prelegerea, dialogul, problematizarea, explicația	

5. Interfețe și excepții - clase abstracte - interfețe - excepții în Java	Prelegerea, dialogul, problematizarea, explicația	
6. Biblioteci de clase - clase învelitoare - structura de dată listă - algoritmi de căutare și sortare	Prelegerea, dialogul, problematizarea, explicația	
7. Operații I/O în Java - fluxuri pentru date binare - fluxuri bazate pe caractere - serializarea obiectelor	Prelegerea, dialogul, problematizarea, explicația	
8. Interfața grafică - ierarhia de clase AWT și Swing - gestionări de poziționare - componente Swing	Prelegerea, dialogul, problematizarea, explicația	
9. Programare bazată pe evenimente - clase de evenimente - model de delegare al evenimentelor	Prelegerea, dialogul, problematizarea, explicația	
10. Colecții - interfețe pentru colecții și implementarea lor - algoritmi și iteratori	Prelegerea, dialogul, problematizarea, explicația	
11. Fire de execuție - crearea firelor de execuție - prioritatea firelor de execuție - interconectarea firelor de execuție	Prelegerea, dialogul, problematizarea, explicația	
12. Șabloane de proiectare - implementarea claselor și metodelor pe baza diagramelor - șablonul creațional (Abstract Factory, Singleton)	Prelegerea, dialogul, problematizarea, explicația	
13. Șabloane de proiectare structurale - șabloanele Bridge, Composite, Decorator, Flyweight și Proxy	Prelegerea, dialogul, problematizarea, explicația	
14. Șabloane de proiectare comportamentale - Command, Iterator, Observer, Strategy și Visitor	Prelegerea, dialogul, problematizarea, explicația	
Bibliografie 1. Eckel, B., Thinking in Java (4th ed.), New York: Prentice Hall, 2006. 2. Eckel, B., Thinking in Patterns with Java, MindView Inc, 2004. 3. Gamma, E., Helm, R., Johnson, R., Vlissides, J., Programtervezési minták. Újrahasznosítható elemek objektumközpontú programokhoz, Kiskapu, Addison-Wesley, 2004. 4. Sierra K., Bates B.: Agyhullám: Java, Kiskapu, O'Reilly, 2011.		
8.2 Seminar / laborator	Metode de predare	Observații
S1. Programe simple în Java	exerciții, muncă individuală, dialog	
S2. Polimorfism	exerciții, muncă individuală, dialog	
S3. Interfețe	exerciții, muncă individuală, dialog	
S4. Excepții	exerciții, muncă individuală, dialog	
S5. Aplicarea șablonului de proiectare Observer	exerciții, muncă individuală, dialog, proiect practic	
S6. Aplicarea șablonelor de proiectare Composite și Decorator	exerciții, muncă individuală, dialog, proiect practic	
S7. Aplicarea șablonelor de proiectare Strategy și Visitor	exerciții, muncă individuală, dialog	
L1. Programe simple în Java	exerciții, muncă individuală, dialog	
L2. Implementarea tipurilor abstracte de date în Java	exerciții, muncă individuală, dialog	

L3-4. Programre bazată pe interfețe	exerciții, muncă individuală, dialog	
L5. Aplicații AWT	exerciții, muncă individuală, dialog	
L6. Aplicații Swing	exerciții, muncă individuală, dialog	
L7. Operații I/O	exerciții, muncă individuală, dialog	
L8. Serializare	exerciții, muncă individuală, dialog	
L9. Aplicații ale șablonului Abstract Factory	exerciții, muncă individuală, dialog	
L10. Utilizarea firelor de execuție	exerciții, muncă individuală, dialog	
L11-12. Aplicații ale șablonului Decorator	exerciții, muncă individuală, dialog	
L13-14. Aplicații ale șablonului Composite	exerciții, muncă individuală, dialog	
Bibliografie 1. Angster E. Objektumorientált tervezés és programozás. Java. I. és II. kötet, 4KÖR Bt., 2003, 2004. 2. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. 3. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2000.		

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

- Conținutul cursului este în concordanță cu cursuri asemănătoare predate în țară sau în străinătate, care se referă la șabloane de proiectare și programe în Java.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Activitate la curs și cunoștințe teoretice	Examen teoretic	40%
10.5 Seminar/laborator	Activitatea la seminar	Exerciții la seminar	10%
	Activitatea la laborator	Exerciții la laborator, teme pentru acasă și proiect	25%
	Însușirea cunoștințelor practice	Examen practic	25%
10.6 Standard minim de performanță			
• Cunoașterea elementelor de bază ale limbajului de programare Java. • Aplicarea șabloanelor pentru probleme de proiectare orientate pe obiecte. • La examenul teoretic și cel practic trebuie obținut cel puțin 50%.			

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

Nu se aplică.

Data completării:

20.09.2025

Semnătura titularului de curs

Conf. univ. dr. Darvay Zsolt

.....

Semnătura titularului de seminar

Conf. univ. dr. Darvay Zsolt

.....

Data avizării în departament:

20.09.2025

Semnătura directorului de departament

Conf. univ. dr. András Szilárd

.....