

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai din Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematică și Informatică</b>
1.3 Departamentul	<b>Departamentul de Informatică</b>
1.4 Domeniul de studii	<b>Matematică-Informatică</b>
1.5 Ciclul de studii	<b>Licență</b>
1.6 Programul de studiu / Calificarea	<b>Matematică-Informatică - limba de studiu română</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Metode Avansate de Programare</b>						
2.2 Titularul activităților de curs	<b>Conf. Univ. Dr. Molnar Arthur</b>						
2.3 Titularul activităților de seminar	<b>Conf. Univ. Dr. Molnar Arthur</b>						
2.4 Anul de studii	<b>2</b>	2.5 Semestrul	<b>3</b>	2.6. Tipul de evaluare	<b>C</b>	2.7 Regimul disciplinei	<b>Obligatorie</b>
2.8 Codul Disciplinei	<b>MLR5008</b>						

### 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	1 Sem + 1 Lab
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					4
Examinări					7
Alte activități: .....					-
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	Algoritmi și Programare, Programare Orientată Obiect, Structuri de Date
4.2 de competențe	Competențe de programare la nivel mediu într-un limbaj de nivel înalt.

### 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	Sală, plus proiector
-------------------------------	----------------------

5.2 De desfășurare a seminarului/laboratorului	Săli de curs/seminar echipate pentru dezvoltare Java.
--	---

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<p>C1.1 Cunoașterea, înțelegerea și utilizarea conceptelor de bază în analiza și proiectarea orientată obiect.</p> <p>C1.2 Abilitatea de a lucra independent sau în echipă pentru rezolvarea unor probleme la scară mică sau medie.</p> <p>C1.3 Competențe legate de dezvoltarea programelor în paradigma orientată obiect folosind limbajul Java.</p> <p>C1.4 Utilizarea șablonelor de proiectare în contexte potrivite.</p> <p>C1.5 Implementarea de soluții informatice bazate pe diferite șabloane arhitecturale.</p>
<b>Competențe transversale</b>	<p>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</p> <p>CT2 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</p>

## 7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Capacitatea de a analiza, proiecta și implementa o soluție informatică de dimensiune mică/medie folosind o interfață grafică cu utilizatorul în limbajul Java.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Utilizarea conceptelor programării orientate obiect în cadrul analizei și proiectării soluției.</li> <li>• Implementarea soluțiilor folosind platforma Java.</li> <li>• Aplicarea șablonelor de proiectare corecte în mai multe contexte.</li> <li>• Integrarea de clase/componente scrise de alți dezvoltatori în propriile creații.</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
<b>1. Introducere în platforma Java</b> (platforma, sintaxa, tipuri de date, vectori, exemple)	<p>Expunere interactivă</p> <p>Explicație</p> <p>Conversație</p> <p>Exemple</p> <p>Demonstrație didactica</p>	-
<b>2. Clase</b> (clase, construirea obiectelor, metode, încapsulare, moștenire, polimorfism, clase abstracte, interfețe)		
<b>3. Tipuri generice, colecții Java</b> (metode generice, ștergerea tipurilor. clase generice și subtipizare, wildcards, Java Collections Framework)		
<b>4. Excepții, I/O în Java, JUnit</b> (ierarhia de excepții Java, fluxuri, serializare, testare unitară folosind JUnit)		
<b>5. JDBC, Elemente de programare funcțională</b> (JDBC API, elemente Java 8 - expresii lambda, fluxuri)		
<b>6. Interfețe grafice cu utilizatorul</b> (platforma JavaFX, scene, modalități de dispunere, controale,		

evenimente, procesarea evenimentelor, șablonul MVC, limbajul FXML)		
<b>7. Reflecție, Concurență</b> (API-ul Java Reflection, concurență - fire de execuție, procese, programarea multi-fir de execuție în Java, sincronizarea firelor de execuție)		
<b>8. Șabloane de Proiectare</b> (șabloane creaționale, structurale, comportamentale)		
<b>9. Pregătirea examenului scris</b> - ghidul de examen		
<b>10. Examinarea scrisă</b> (colocviu)		
<b>Bibliografie</b>		
<ol style="list-style-type: none"> <li>1. Eckel, B. Thinking in Java, 4th edition, Prentice Hall, 2006.</li> <li>2. Eckel, B. Thinking in Patterns with Java, 2004. MindView, Inc.</li> <li>3. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Longman Publishing, 1995.</li> <li>4. The Java Tutorials: <a href="https://docs.oracle.com/javase/tutorial/">https://docs.oracle.com/javase/tutorial/</a></li> <li>5. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilly, 2010.</li> </ol>		

<b>8.2 Seminar</b>	Metode de predare	Observații
1. Probleme simple în Java. Arhitectura stratificată (recapitulare)	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactica</li> </ul>	-
2. Moștenire, interfețe, pachete, iteratori.		
3. Tipuri generice, colecții, excepții.		
4. Serializarea, lucrul cu fișiere, JDBC		
5. Interfețe grafice cu utilizatorul prin JavaFX		
6. Fire de execuție, programare concurentă		
7. Șabloane de proiectare		
<b>Bibliografie</b> - similară cursului.		

<b>8.3 Laborator</b>	Metode de predare	Observații
1. Instalarea/configurarea JDK, a unui IDE, lucrul cu sistemul git. Probleme introductive în Java	<ul style="list-style-type: none"> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> </ul>	-
2. Arhitectura stratificată, tipuri generice, excepții.		
3. Fișiere, serializare, JUnit		
4. JDBC, elemente de programare funcțională		
5. Test practic în timpul laboratorului		
6. Interfețe grafice cu utilizatorul		
7. Examen practic		
<b>Bibliografie</b> - similară cursului.		

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

Cursul respectă curriculumul IEEE și ACM pentru domeniul Informatică.

Cursul există în programele de studiu ale universităților importante din România și din străinătate.

Conținutul disciplinei este considerat de majoritatea companiilor software ca fiind deosebit de important pentru obținerea unor abilități avansate de programare.

## **10. Evaluare**

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	Cunoștințele legate de analiza și proiectarea sistemelor software, precum și aplicarea principiilor deprinse în dezvoltarea de aplicații folosind platforma Java.	Colocviu sub forma unui test scris susținut în cadrul ultimului curs.	<b>30%</b>
10.5 Seminar/laborator	Competențe legate de implementarea unor soluții informatice de dimensiune mică/medie folosind platforma Java	Evaluare continuă pe durata semestrului, pe parcursul orelor de laborator.	<b>40%</b>
10.5 Seminar/laborator		Colocviu sub forma unui test practic susținut în cadrul ultimei ore de laborator.	<b>30%</b>
<b>10.6 Standard minim de performanță</b>			
<ul style="list-style-type: none"> <li>• Studenții vor respecta standardele de integritate academică. Temele de laborator vor fi implementate în mod independent, și pentru a fi notate studenții vor demonstra o bună înțelegere a codului sursă precum și a conceptelor utilizate în cadrul implementării.</li> <li>• Studenții trebuie să demonstreze o bună înțelegere a conceptelor prezentate și exersate în timpul cursului</li> <li>• Pentru a intra în procesul de evaluare (colocviu desfășurat pe durata semestrului), studenții trebuie să îndeplinească următoarele cerințe legate de prezența la ore: <u>minimum 4 prezențe în cadrul seminarului și 5 prezențe în cadrul laboratorului</u>, pe durata primelor 12 săptămâni de activitate didactică.</li> <li>• Examenul va fi considerat promovat în condiția obținerii unei <u>note mai mari sau egale cu 5.00</u> concomitent în cadrul testului scris (colocviu), cât și al celui practic (colocviu).</li> </ul>			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

Conf. Univ Dr. Molnar Arthur

Conf. Univ. Dr. Molnar Arthur

Data avizării în departament

Semnătura directorului de departament

Conf. Univ. Dr. Sterca Adrian