

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai din Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Informatica
1.5 Ciclul de studii	Postuniversitar
1.6 Programul de studiu / Calificarea	Program postuniversitar de formare si dezvoltare profesionala in Informatica

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)	Programare orientată obiect						
2.2 Titularul activităților de curs	Lect. Dr. Simon Károly						
2.3 Titularul activităților de seminar	Lect. Dr. Simon Károly						
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie
2.8 Codul disciplinei	MLM5122						

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	2
3.4 Total ore din planul de învățământ	40	Din care: 3.5 curs	20	3.6 seminar/laborator	20
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					30
Tutoriat					6
Examinări					4
Alte activități:					0
3.7 Total ore studiu individual		85			
3.8 Total ore pe semestru		125			
3.9 Numărul de credite		5			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Fundamentele Programarii si Algoritmica, Programare si Structuri de Date
4.2 de competențe	<ul style="list-style-type: none"> Cunoștințe medii de programare într-un limbaj de programare de nivel înalt

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> Sală, plus proiector
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Sală de laborator cu calculatoare dotate cu limbajul de programare Java

6. Competențele specifice acumulate

Competențe profesionale	<p>C1.1 Descrierea adecvată a paradigmatelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</p> <p>C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</p> <p>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</p> <p>C1.4 Testarea unor aplicații pe baza unor planuri de test</p> <p>C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente</p>
Competențe transversale	<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>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</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> Să deprindă studentul cu proiectarea orientată obiect a problemelor de scară mică/mijlocie și învățarea limbajului de programare Java, precum și cu deprinderea principiilor unei bune proiectării orientate obiect
7.2 Obiectivele specifice	<ul style="list-style-type: none"> Să cunoască conceptele de baza ale programării Să cunoască conceptele de baza ale ingineriei software Să folosească instrumente de baza pentru construirea programelor Să învețe limbajul Java și instrumente de dezvoltare pentru programarea, executia și depanarea programelor Java. Să-și însușească principiile unei bune proiectări orientate obiect precum și a sabloanelor de proiectare prezentate Să-și însușească un stil de programare conform celor mai bune recomandări practice.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere în platforma Java: istoria și proprietățile limbajului, sintaxa, compilarea și rularea programelor, medii de dezvoltare, exemple/programe simple.	<ul style="list-style-type: none"> Expunere interactivă Explicație Conversație Exemple Demonstrație didactică 	
2. Elemente de bază a limbajului Java: tipuri de date primitive, tipuri de referință (clase	<ul style="list-style-type: none"> Expunere interactivă Explicație 	

simple, clase wrapper), tablouri, excepții, pachete	<ul style="list-style-type: none"> • Conversație • Exemple • Demonstrație didactică 	
3. POO – Concepte de bază: tip abstract de date (TAD), clasă, obiect, starea unui obiect, relații între clase și obiecte, proprietăți și metode, constructori, diagrame de clase (UML)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
4. Modificatori de acces (vizibilitatea membrilor), modificatori de tip	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
5. Moștenire și polimorfism: clase derivate, redefinirea metodelor, principiul substituției, polimorfism	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
6. Clase abstracte și interfețe: noțiunea de clasă abstractă, noțiunea de interfață (în Java), rețete generale legate de utilizarea interfețelor, clase interne (și clase interne anonime)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
7. Java Collections Framework și recapitularea conceptelor învățate prin exemple legate de colecții	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
8. Tipuri generice (tipuri generice și tipuri concrete, mecanismul type erasure în Java, metode generice), adnotări (adnotări simple și meta-adnotări)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
9. Clasa Object, clasa Class și bazele mecanismului reflection (în Java)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
10. Expresii lambda, tehnologia Streams API (+metode statice și metode default la nivelul interfețelor)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
Bibliografie		

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition.
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, Ed. Addison Wesley, 1994
5. ***, The Java Tutorial, 2013. <http://download.oracle.com/javase/tutorial/>
6. Simon Károly, Kenyerünk Java, Egyetemi Kiadó, 2010

8.2 Seminar / laborator	Metode de predare	Observații
1. Java Basics	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
2. Aplicații simple	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
3. Clase și obiecte	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
4. Modificatori de acces și modificatori de tip	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
5. Moștenire și polimorfism	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
6. Clase abstracte și interfețe, clase interne	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
7. Java Collections Framework	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
8. Generics	<ul style="list-style-type: none"> • Expunere interactivă • Explicație 	

	<ul style="list-style-type: none"> • Conversație • Exemple • Demonstrație didactică 	
9. Reflection	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
10. Expresii lambda și Streams API	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
Bibliografie 1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition. 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, Ed. Addison Wesley, 1994 5. ***, The Java Tutorial, 2013. http://download.oracle.com/javase/tutorial/ 6. Simon Károly, Kenyerünk Java, Egyetemi Kiadó, 2010		

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

<ul style="list-style-type: none"> • Cursul respecta recomandările IEEE și ACM legate de Curricula pentru specializarea Informatică. • Cursul face parte din programul de studiu de la majoritatea universităților importante din România și din străinătate. • Conținutul cursului este considerat de companiile soft ca fiind important pentru un nivel mediu de cunoștințe în programare.

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. Capacitatea de a proiecta și implementa programe scrise în limbajul Java	Examen scris	50%
10.5 Seminar/laborator	Abilitatea de a scrie și depana un program Java	Examen practic	50%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Minimum 5 la fiecare proba. 			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

Lect. Dr. Simon Károly

Lect. Dr. Simon Károly

Data avizării în departament

Semnătura directorului de departament

Conf. dr. András Szilárd Károly