

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica și Informatică
1.3 Departamentul	Departamentul de matematică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Metode avansate de programare						
2.2 Titularul activităților de curs	Lect. univ. dr. Cioban Vasile						
2.3 Titularul activităților de seminar	Lect. univ. dr. Cioban Vasile						
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	5	Din care: 3.2 curs	2	3.3 seminar/laborator	1 sem / 2 lab
3.4 Total ore din planul de învățământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	42
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					30
Tutoriat					10
Examinări					10
Alte activități:					-
3.7 Total ore studiu individual		80			
3.8 Total ore pe semestru		150			
3.9 Numărul de credite		6			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> • Programare orientată obiect • Structuri de date și algoritmi
4.2 de competențe	<ul style="list-style-type: none"> • Conceptele de bază ale programării orientate obiect

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	•
5.2 De desfășurare a seminarului/laboratorului	• Laboratoare pe calculator

6. Competențele specifice acumulate

Competențe profesionale	<p>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.</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"> • Înțelegerea și folosirea conceptelor programării orientate obiect la dezvoltarea unor aplicații soft de complexitate medie • Înțelegerea și folosirea unor șabloane de proiectare pentru dezvoltarea aplicațiilor. • Deprinderea dezvoltării unor aplicații cu interfețe grafice • Deprinderea folosirii unor frameworkuri pentru dezvoltarea aplicațiilor soft.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Consolidarea deprinderilor de analiză și proiectare a sistemelor soft prin învățarea unui proces liniar simplu de analiză și proiectare orientată obiect. • Însușirea unor elemente de bază ale limbajului de modelare UML. • Însușirea unor șabloane de proiectare care se folosesc pentru structurarea logică a aplicațiilor. • Aplicarea unor șabloane standard privind alocarea responsabilităților pentru obiecte. • Învățarea limbajului Java. • Învățarea limbajului C# • Însușirea unor cunoștințe de bază pentru crearea programelor cu interfețe grafice utilizator.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
C1 . Introducere în Java.	Expuneri: concepte,	

Introducere in C#	exemple, studii de caz	
C2. Clase si obiecte Java / C#	Expuneri: concepte, exemple, studii de caz	
C3. Mostenire Clase abstracte - Template Method Design & Adaptor Patterns Interfete si implementarea interfetelor	Expuneri: concepte, exemple, studii de caz	
C4. Tratarea exceptiilor Generics	Expuneri: concepte, exemple, studii de caz	
C5. Pachete, modificatori de acces Clase interioare Biblioteca cu structuri de date java.util	Expuneri: concepte, exemple, studii de caz	
C6. Biblioteci pentru intrari-iesiri Proiectarea unei aplicatii pe 3 nivele	Expuneri: concepte, exemple, studii de caz	
C7-C8. Construirea interfetelor grafice utilizator Tratarea evenimentelor.	Expuneri: concepte, exemple, studii de caz	
C9. Sablone de proiectare. Sabloane GRASP MVC: model-prezentare-control, control pentru fereastra si control pentru aplicatie	Expuneri: concepte, exemple, studii de caz	
C10. Aplicatiile sablonului Observer, Validarea datelor.	Expuneri: concepte, exemple, studii de caz	
C11 . Introducere in analiza si proiectarea orientata obiect (determinarea cerintelor, cazuri de utilizare, analiza, proiectare).	Expuneri: concepte, exemple, studii de caz	
C12. Persistenta obiectelor	Expuneri: concepte, exemple, studii de caz	
C13. Factory patterns. Dependency Injection. Inversion of Control	Expuneri: concepte, exemple, studii de caz	
C14. Reflection	Expuneri: concepte, exemple, studii de caz	

<http://www.cs.ubbcluj.ro/~vniculescu/didactic/>

Bibliografie

1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.
2. Eckel, B., Thinking in C#, MindView Inc, 2004.
3. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.
4. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
5. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.
6. Walls, Craig, Spring in Action, Third Edition, Ed. O'Reilley, 2011.
7. Kent Beck, Test Driven Development: By Example, Ed. Addison-Wesley Professional, 2002.

8. ***, Tutoriale Java http://download.oracle.com/javase/tutorial/		
9. ***, Tutoriale C# http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx		
8.2 Seminar	Metode de predare	Observații
1. Programe simple Java / C#	Discutii, exemple, studii de caz	Seminarul este de 2 ore o data la 2 saptamani
2. Polimorfism . reutilizare	Discutii, exemple, studii de caz	
3. I/O. Studii de caz	Discutii, exemple, studii de caz	
4. Interfete grafice	Discutii, exemple, studii de caz	
5. Sablonul Observer. Studiu de caz	Discutii, exemple, studii de caz	
6. Sabloane de proiectare	Discutii, exemple, studii de caz	
7. Proiectarea si implementarea nivelului de persistenta	Discutii, exemple, studii de caz	
8.3 Laborator		
L1. Program simplu Java/ C#	Discutii, evaluare	
L2-L4. Unit Testing	Discutii, evaluare	
L5. Tratarea exceptiilor	Discutii, evaluare	
L6. I/O	Discutii, evaluare	
L7. Interfete grafice simple	Discutii, evaluare	
L8-L10. Aplicatie de complexitate medie cu interfata grafica.	Discutii, evaluare	
L11-L12. Folosirea sablonului Observer	Discutii, evaluare	
L13-L14. Dezvoltarea unei aplicatii de complexitate	Discutii, evaluare	
Bibliografie		
10. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.		
11. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.		
12. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.		
13. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.		
14. Walls, Craig, Spring in Action, Third Edition, Ed. O’Reilley, 2011.		
15. Kent Beck, Test Driven Development: By Example, Ed. Addison-Wesley Professional, 2002.		
16. ***, Tutoriale Java http://download.oracle.com/javase/tutorial/		
17. ***, Tutoriale C# http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx		

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 curriculei ACM si IEEE pentru studiile de informatica. • Cursul apare in planurile de invatamant a celor mai importante universitati din tara si strainatate. • Firmele de soft considera continutul cursului important pentru dobandirea unor abilitati medii de programare.

10. Evaluare

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

			nota finală
	Cunoasterea conceptelor de baza	Examen partial	20%
10.4 Curs	Cunoasterea conceptelor de baza	Examen scris	15%
	Aplicarea conceptelor de baza	Examen practic	25%
	Folosirea conceptelor introduse la curs pentru rezolvarea unor probleme concrete	Evaluare proiect	15%
10.5 Seminar/laborator	Folosirea conceptelor introduse la curs pentru rezolvarea unor probleme concrete	Evaluarea calitatii si completitudinii temelor primite	25%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Pe o scara de la 1 la 10: <ul style="list-style-type: none"> ○ Notele la laborator, partial si examen cel putin 4. ○ Media finala trebuie sa fie cel putin 5 (pe o scara de la 1 la 10). 			

Data completării

... 15.05.2016.....

Titular de curs

Lect. univ. dr. Cioban Vasile

Titular de seminar

Lect. univ. dr. Cioban Vasile

Data avizării în departament

.....

Director de departament

.....