

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematica și Informatică</b>
1.3 Departamentul	<b>Departamentul de Informatică</b>
1.4 Domeniul de studii	Informatica
1.5 Ciclul de studii	Postuniversitar
1.6 Programul de studiu / Calificarea	<b>Program postuniversitar de informatică și dezvoltare software (în limba maghiară)</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Proiect colectiv</b>						
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	<b>2</b>	2.5 Semestrul	<b>3</b>	2.6. Tipul de evaluare	<b>VP</b>	2.7 Regimul disciplinei	<b>obligatoriu</b>

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

3.1 Număr de ore pe săptămână	10	Din care: 3.2 curs	0	3.3 seminar/laborator	10
3.4 Total ore din planul de învățământ	100	Din care: 3.5 curs	0	3.6 seminar/laborator	100
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					80
Tutoriat					30
Examinări					5
Alte activități: .....					-
3.7 Total ore studiu individual	150				
3.8 Total ore pe semestru	250				
3.9 Numărul de credite	10				

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> <li>Fundamentele Programării și Algoritmica, Programare și Structuri de Date, Programare Orientată Obiect, Metode și Medii avansate de programare, Baze de date, Software Engineering și șabloane de proiectare</li> </ul>
4.2 de competențe	<ul style="list-style-type: none"> <li>Cunoștințe de programare într-un limbaj de programare de nivel înalt, orientat obiect</li> </ul>

## 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	
5.2 De desfășurare a seminarului/laboratorului	Activitatea trebuie să includă activități de specialitate: programare, testare, analiza și proiectare sisteme software

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	C2.1 Identificarea de metodologii adecvate de dezvoltare a sistemelor software. C2.2 Identificarea și explicarea mecanismelor adecvate de specificare a sistemelor software.
<b>Competențe transversale</b>	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. CT2 Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatice de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse. 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

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

7.1 Obiectivul general al disciplinei	- Competențe îmbunătățite de proiectare și programare
7.2 Obiectivele specifice	- Familiarizarea cu concepte moderne din dezvoltarea software - Realizarea unui proiect de un grup de studenți - Elaborarea documentațiilor necesare - Prezentarea proiectului

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
8.2 Seminar / laborator	Metode de predare	Observații

1. Discutii generale legate de activitatile de la laborator. Prezentarea modului de lucru, stabilirea cerintelor. Prezentarea temelor, repartizarea proiectelor. Stabilirea termenelor de predare.	Explicare, exemplificare, dezbateri, dialog	
2. Definire planului de proiect, specificații	Explicare, exemplificare, dezbateri, dialog	
3. Analiza proiectului - Use case analysis, domain analysis	Explicare, exemplificare, dezbateri, dialog	
4. Plan de proiect - Architecture design - Identificarea/selecția tehnologiilor folosite	Explicare, exemplificare, dezbateri, dialog	
5. Detaliere plan proiect.	Explicare, exemplificare, dezbateri, dialog.	
6. Implementare	Explicare, exemplificare, dezbateri, dialog, evaluare.	
7. Testare, code review	Explicare, exemplificare, dezbateri, dialog.	
8. Prezentarea proiectelor	Evaluare.	
Referinte: 1. Eckel, B.: Thinking in Patterns. Problem-Solving Techniques using Java ( <a href="http://www.bruceeckel.com">http://www.bruceeckel.com</a> ). 2. Fowler M.: Patterns of Enterprise Application Architecture, Addison-Wesley, Pearson Education, 2003. 3. Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995 4. Alte surse specifice proiectului		

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

Cursul respecta recomandările IEEE și ACM legate de Curricula pentru specializarea Informatică

- Cursul ofera o imagine de ansamblu asupra mai multor domenii din Informatica, ofera studentului o expertiza generala asupra Informaticii.
- Cursul ofera cunostinte de baza despre lucrul în echipă și integrare în piața muncii

## 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs			
Seminar/laborator		Prezentare, evaluarea codului sursă și a documentației	50%
		Verificare pe parcurs, evaluarea activităților de dezvoltare	50%
Standard minim de performanță			
• Minimum 5 la fiecare proba			

Data completării

30.08.2020

Semnătura titularului de curs

Lect. dr. Simon Károly

Semnătura titularului de seminar

Lect. dr. Simon Károly

Data avizării în departament

.....

Semnătura directorului de departament

Conf. dr. András Szilárd