

FIȘA DISCIPLINEI

Ingineria sistemelor soft

Anul universitar 2024-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babes Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licenta
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Ingineria sistemelor soft				Codul disciplinei		MLE5011			
2.2. Titularul activităților de curs			Lect. Dr. Zsigmond Imre								
2.3. Titularul activităților de seminar			Lect. Dr. Zsigmond Imre								
2.4. Anul de studiu		2	2.5. Semestrul		2	2.6. Tipul de evaluare		C	2.7. Regimul disciplinei		DF

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/proiect	3
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					24
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					24
Tutoriat (consiliere profesională)					10
Examinări					10
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				80	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Metode avansate de programare
4.2. de competențe	Abilități medii de programare într-un limbaj de programare de nivel înalt orientat pe obiecte

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Proiector
5.2. de desfășurare a seminarului/ laboratorului	Proiector

6.1. Competențele specifice acumulate¹

Competențe profesionale/esențiale	- C2.3 - Abilitatea de a lucra independent și în echipă în vederea dezvoltării de software care respectă standardele industriale. - C2.5 - Înțelegerea rolului diferitelor artefacte utilizate în procesul de dezvoltare a software-ului și dobândirea capacității de realizare și utilizare a acestor artefacte
Competențe transversale	- CT2 - Abilitatea de a crea software începând cu construcția modelului, continuând cu verificarea modelului și transformarea modelului în cod, realizând și utilizând modele de testare - CT3 - Abilitatea de a utiliza o metodologie software pentru a produce software de calitate, de la analiza cerințelor software până la generarea de cod și testarea software-ului

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: tehnici standard din industrie pentru proiectarea, implementarea, testarea și întreținerea sistemelor software complexe. Cunoștințele dobândite includ ciclul de viață al unui proiect, un număr redus de diagrame UML, colectarea cerințelor; lucrul cu sarcini, utilizarea Git în echipă, cunoștințe extinse de C# cu .NET, dezvoltare desktop cu WinUI și dezvoltare web cu ASP.NET (MVC și API), utilizarea Entity Framework, teste unitare obișnuite și simulate, mai multe tipare de proiectare și tipare arhitecturale.
Aptitudini	Studentul este capabil să lucreze în echipe de dimensiuni mici până la medii (6–30 de membri). Studentul a dobândit sau și-a dezvoltat următoarele competențe: ingineria cerințelor; UML, lucrul în echipă, Git, interfețe grafice/design, arhitecturi software, C# + .NET, calitatea codului, abilități de comunicare, testare unitară, acces la baze de date, dezvoltare web, tipare de proiectare, implementare (deployment), cod curat.
Responsabilități și autonomie	Studentul are capacitatea de a lucra în echipă pentru a crea software de calitate.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	- Să fie capabil să înțeleagă ciclul de viață al producției de software - Abilități îmbunătățite în dezvoltarea de software	•
7.2 Obiectivele specifice	- Să fie capabil să dezvolte software în echipă - Înțelegeți cele mai bune practici implementate în industria software - Să fie capabil să comunice mai bine cu ceilalți în chestiuni tehnice - Înțelegeți diferite arhitecturi	•

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere în ingineria software	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
2. Design de software	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
3. Lucrul în echipă	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
4. C# + .NET with WinUI	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
5. Calitatea codului	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
6. Unit testing	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
7. Design patterns	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
8. Managementul dependenței	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
9. ASP.NET MVC	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
10. Servere	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
11. Proiecte software și managementul lor	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
12. Arhitecturi software	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
13. Angajare	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
14. Examen		
Bibliografie 1. Andrew Troelsen, Phil Japikse: Pro C# 10 with .NET 6 2. Robert C. Martin: Clean code 3. Robert C. Martin: Clean architecture 4. Roy Osherove: The art of unit testing 5. Scott Chacon: Pro Git 6. Martin Fowler: Patterns of Enterprise Application Architecture 7. Bruce M. Van Horn II: Real-World Implementation of C# Design Patterns 8. Adam Freeman: Pro ASP.NET Core 6 9. Konstantin Semenenko: C# Interview Guide		
8.2 Seminar	Metode de predare	Observații
1. Colectarea cerințelor	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
2. Prototipe pe hartie	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
3. Dezvoltare cu GIT în echipe	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
4. Folosirea IA în dezvoltare	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
5. Instalarea aplicației pe un server	Expunere: descriere, explicație, exemple, discuții despre studii de caz	

6. Generate de code	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
7. Support de proiect	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
8.3 Laborator	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
1. Configurarea mediului și C#	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
2. Planificarea versiunii inițiale a proiectului cu utilizarea uml și tehnici de management de proiect	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
3. Dezvoltare software în echipe	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
4. Revizuirea codului, refactorizarea, testarea unitară și de integrare	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
5. Dezvoltare client-server in echipe mari	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
6. Dezvoltare web	Expunere: descriere, explicație, exemple, discuții despre studii de caz	
7. Suport multi-platformă în echipe și mai mari	Expunere: descriere, explicație, exemple, discuții despre studii de caz	

Bibliografie

1. Andrew Troelsen, Phil Japikse: Pro C# 10 with .NET 6
2. Robert C. Martin: Clean code
3. Robert C. Martin: Clean architecture
4. Roy Oshero: The art of unit testing
5. Scott Chacon: Pro Git
6. Martin Fowler: Patterns of Enterprise Application Architecture
7. Bruce M. Van Horn II: Real-World Implementation of C# Design Patterns
8. Adam Freeman: Pro ASP.NET Core 6
9. Konstantin Semenenko: C# Interview Guide

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

- Cursul respectă Recomandările IEEE și ACM pentru Curricula pentru Studii în Informatică;
- Cursul există în programul de studii al tuturor universităților importante din România și din străinătate;
- Conținutul cursului conține cunoștințe obligatorii pentru orice specialist IT care lucrează într-o companie de software

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașteți conceptele prezentate și principiile SE	Proiect in echipa	0%
10.5 Seminar/laborator	Să fie capabil să implementeze cunoștințe recunoscute în producerea de software	Proiect in echipa	100%

10.6 Standard minim de performanță
•

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:
01/05/2025

Semnătura titularului de curs



Semnătura titularului de seminar



Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică*".