

FIȘA DISCIPLINEI

Anul universitar 2023 - 2024

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Lucian Blaga din Sibiu
1.2. Facultatea	Științe
1.3. Departament	Departamentul de Matematică și Informatică
1.4. Domeniul de studiu	Informatică
1.5. Ciclul de studii ¹	Master
1.6. Specializarea	Sisteme și tehnologii informatice avansate

2. Date despre disciplină

2.1. Denumirea disciplinei	Medii și platforme de dezvoltare avansate	Co d	FSTI.MAI.STIA.M.SO. 2.2020.E-7.1
2.2. Titular activității de curs	Lector univ. dr. Hunyadi Daniel		
2.3. Titular activității practice	Lector univ. dr. Hunyadi Daniel		
2.4. An de studiu ²	1	2.5. Semestrul ³	2
2.6. Tipul de evaluare ⁴			E
2.7. Regimul disciplinei ⁵	O	2.8. Categoria formativă a disciplinei ⁶	S

3. Timpul total estimat

3.1. Extinderea disciplinei în planul de învățământ – număr de ore pe săptămână					
3.1.a.Curs	3.1.b. Seminar	3.1.c. Laborator	3.1.d. Proiect	3.1.e Alte	Total
2		2			4
3.2. Extinderea disciplinei în planul de învățământ – total ore din planul de învățământ					
3.2.a.Curs	3.2.b. Seminar	3.2.c. Laborator	3.2.d. Proiect	3.2.e Alte	Total ⁷
28		28			56
Distribuția fondului de timp pentru studiu individual⁸					Nr. ore
Studiul după manual, suport de curs, bibliografie și notițe					40
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					43

¹ Licență / Master

² 1-4 pentru licență, 1-2 pentru master

³ 1-8 pentru licență, 1-3 pentru master

⁴ Examen, colocviu sau VP A/R – din planul de învățământ

⁵ Regim disciplină: O=Disciplină obligatorie; A=Disciplină opțională; U=Facultativă

⁶ Categoria formativă: S=Specialitate; F=Fundamentală; C=Complementară; I=Asistată integral; P=Asistată parțial; N=Neasistată

⁷ Este egal cu 14 săptămâni x numărul de ore de la punctul 3.1 (similar pentru 3.2.a.b.c.d.e.)

⁸ Liniile de mai jos se referă la studiul individual; totalul se completează la punctul 3.37.

Tutoriat ⁹	14
Examinări ¹⁰	2
3.3. Total ore alocate studiului individual¹¹ (NOSI_{sem})	119
3.4. Total ore din Planul de învățământ (NOAD_{sem})	56
3.5. Total ore pe semestru¹² (NOAD_{sem} + NOSI_{sem})	175
3.6. Nr ore / ECTS	25
3.7. Număr de credite¹³	7

⁹ Între 7 și 14 ore

¹⁰ Între 2 și 6 ore

¹¹ Suma valorilor de pe liniile anterioare, care se referă la studiul individual.

¹² Suma (3.5.) dintre numărul de ore de activitate didactică directă (NOAD) și numărul de ore de studiu individual (NOSI) trebuie să fie egală cu numărul de credite alocate disciplinei (punctul 3.7) x nr. ore pe credit (3.6.)

¹³ Numărul de credit se calculează după formula următoare și se rotunjește la valori vecine întregi (fie prin micșorare fie prin majorare)

$$\text{Nr.credite} = \text{NOCpSpD} \times \text{CC} + \text{NOApSpD} \times \text{CATOCpSdP} \times \text{CC} + \text{TOApSdP} \times \text{CA} \times 30 \text{ credite}$$

Unde:

- NOCpSpD = Număr ore curs/săptămână/disciplina pentru care se calculează creditele
- NOApSpD = Număr ore aplicații (sem./lab./pro.)/săptămână/disciplina pentru care se calculează creditele
- TOCpSdP = Număr total ore curs/săptămână din plan
- TOApSdP = Număr total ore aplicații (sem./lab./pro.)/săptămână din plan
- C_C/C_A = Coeficienți curs/aplicații calculate conform tabelului

Coeficienți

Curs

Aplicații (S/L/P)

Licență

2

1

Master

2,5

1,5

Licență lb. străină

2,5

1,25



4. Precondiții (acolo unde este cazul)

4.1. Discipline necesar a fi promovate anterior (de curriculum) ¹⁴	-
4.2. Competențe	-

5. Condiții (acolo unde este cazul)

5.1. De desfășurare a cursului ¹⁵	Sală de curs, dotată cu tablă, calculator, videoproiector
5.2. De desfășurare a activităților practice (lab/sem/pr/alte) ¹⁶	Sală de laborator dotată cu calculatoare, software (Visual Studio, SQL Server), coonexiune internet

6. Competențe specifice acumulate¹⁷

		Număr de credite alocate disciplinei ¹⁸	7	Repartizare credite pe competențe ¹⁹
6.1. Competențe profesionale	CP1	Cunoasterea teoretică și capacitatea de a aplica principiile robuste de programare		1
	CP2	Cunoasterea și utilizarea adecvată a metodelor de refactoring pentru a face codul aplicațiilor mai ușor de menținut, mai rezistent și mai sigur		1
	CP3	Cunoasterea teoretică și aplicarea practică a modelelor de design în programare		1
	CP4	Capacitatea de a proiecta module complexe care aplică principiile robuste și modelele de design		1
6.2. Competențe transversale	CT1	Dezvoltarea atitudinii pozitive față de muncă și responsabilitate pentru propria pregătire profesională.		1
	CT2	Dezvoltarea spiritului de muncă în echipă.		1
	CT3	Capacitatea de a pune accent pe rezultate și pe obținerea de produse funcționale		1

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

7.1. Obiectivul general	<ul style="list-style-type: none"> Însușirea noțiunilor generale despre principiile robuste și modelele de design în programare Formarea unor deprinderi de lucru care aplică aceste principii și modele
7.2. Obiectivele specifice	<ul style="list-style-type: none"> Programarea unor teme pentru aprofundarea principiilor robuste și a modelelor de design

8. Conținuturi

8.1. Curs²⁰	Metode de predare²¹	Nr. ore
-------------------------------	---------------------------------------	----------------

¹⁴ Se menționează disciplinele obligatoriu a fi promovate anterior sau echivalente

¹⁵ Tablă, videoproiector, flipchart, materiale didactice specifice, platforme on-line etc.

¹⁶ Tehnică de calcul, pachete software, standuri experimentale, platforme on-line etc.

¹⁷ Competențele din Grilele aferente descrierii programului de studii, adaptate la specificul disciplinei

¹⁸ Din planul de învățământ

¹⁹ Creditele alocate disciplinei se distribuie pe competențe profesionale și transversale în funcție de specificul disciplinei

²⁰ Titluri de capitole și paragrafe

²¹ Expunere, prelegere, prezentare la tablă a problematicii studiate, utilizare videoproiector, discuții cu studenții (pentru fiecare capitol, dacă este cazul)



Recapitularea conceptelor de baza in programarea orientata pe obiect. Incapsulare, mostenire, polimorfism. Clase si instante. Static.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Recapitularea conceptelor avansate in programarea orientata pe obiect. Interfete si clase abstracte. Constructori. Mostenire versus compozitie.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Introducere in fundamentele tehnicilor de programare. Motivatia si nevoile pe care le rezolva. Prezentarea terminologiei folosita in dezvoltarea de software.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Introducerea conceptului de “clean code”. “Refactoring” si “Software design”. “Clean code” din punctul de vedere al expertilor din domeniu.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Tehnici de identificare a problemelor din cod. Identificarea greselilor: comentarii si denumirea variabilelor. Probleme generale.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Identificarea problemelor din cod: metode, clase, design	Prelegere, utilizare videoproiector, discuții cu studenții	2
Denumirea robusta a variabilelor. Scrierea comentariilor. Metafora ziarului.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Standarde de codare. Standarde recunoscute in industrie	Prelegere, utilizare videoproiector, discuții cu studenții	2
Modele de design. Modele creationale, structurale si comportamentale	Prelegere, utilizare videoproiector, discuții cu studenții	2
Modele de design arhitecturale - MVC, MWVM. Repository and unit of work. Fluent interface.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Principii de programare. Principiile SOLID. Principiul responsabilitatii unice. Principiul deschis / inchis.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Principiul substitutiei Liskov. Principiul segregarii interfetelor. Principiul inversiunii de control.	Prelegere, utilizare videoproiector, discuții cu studenții	2
Separarea intereselor in programare. Simplitatea in programare. Metodele DRY si YAGNI	Prelegere, utilizare videoproiector, discuții cu studenții	2
Principii GRASP. Programare defensiva	Prelegere, utilizare videoproiector, discuții cu studenții	2
Total ore curs:		28
8.2. Activități practice (8.2.a. Seminar ²² / 8.2.b. Laborator ²³ / 8.2.c. Proiect ²⁴ / 8.2.d. Alte act.practice ²⁵)	Metode de predare	Nr. ore
Incapsulare, moștenire, polimorfism	Utilizare videoproiector, discuții cu studenții	2
Source control. Exemple: Git & VSTS. Continuous integration	Utilizare videoproiector, discuții cu studenții	2
Exemple de cod. Refactoring	Utilizare videoproiector, discuții cu studenții	2
Alte exemple OOP. Mostenire vs compozitie	Utilizare videoproiector, discuții cu studenții	2
Code smells	Utilizare videoproiector, discuții cu studenții	2
Code smells	Utilizare videoproiector, discuții cu studenții	2

²² Discuții, dezbateri, prezentare și/sau analiză de lucrări, rezolvare de exerciții și probleme etc.

²³ Demonstrație practică, exercițiu, experiment etc.

²⁴ Studiu de caz, demonstrație, exercițiu, analiza erorilor etc.

²⁵ Alte tipuri de activități practice specifice

Coding guidelines	Utilizare videoproiector, discuții cu studenții	2
Coding guidelines	Utilizare videoproiector, discuții cu studenții	2
Design patterns	Utilizare videoproiector, discuții cu studenții	2
Design patterns	Utilizare videoproiector, discuții cu studenții	2
Design patterns	Utilizare videoproiector, discuții cu studenții	2
SOLID	Utilizare videoproiector, discuții cu studenții	2
Dependency injection	Utilizare videoproiector, discuții cu studenții	2
Coupling, cohesion, DRY	Utilizare videoproiector, discuții cu studenții	2
Total ore seminar/laborator:		28

9. Bibliografie

9.1. Referințe bibliografice recomandate	Valer Roșca, Daniel Hunyadi, Realizarea aplicațiilor Web în mediul ASP.NET, Editura Universității “Lucian Blaga” din Sibiu, 2014
	Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009
	Martin Fowler, Refactoring: Improving the Design of Existing Code: Object Technology Series, ADDISON-WESLEY, 2019
	Erich Gamma, Richard Helm, Ralph E. Johnson, John Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994
9.2. Referințe bibliografice suplimentare	https://en.wikipedia.org/wiki/Design_Patterns
	https://en.wikipedia.org/wiki/Software_design_pattern
	http://www.dofactory.com/net/design-patterns
	https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm

10. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului²⁶

Se realizează prin contacte periodice cu reprezentanții firmelor de profil. Dezvoltarea aplicațiilor respectând principiile fundamentale de programare (SOLID, DRY, YAGNI) și implementând design pattern-urile corecte au ca rezultat obținerea de proiecte robuste și performante.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare		11.3 Pondere din nota finală	Obs. ²⁷
11.4a Examen / Colocviu	● Cunoștințe teoretice și practice însușite (cantitatea, corectitudinea, acuratețea)	Teste pe parcurs ²⁸ :	%	10% (minim 5)	CEF
		Teme de casă:	%		
		Alte activități ²⁹ :	%		
		Evaluare finală: 1	10%		

²⁶ Legătura cu alte discipline, utilitatea disciplinei pe piața muncii

²⁷ CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală; N/A – nu se aplică

²⁸ Se va preciza numărul de teste și săptămânile în care vor fi susținute.

²⁹ Cercuri științifice, concursuri profesionale etc.

