

Anexa 2.

## FIȘA DISCIPLINEI\*

### 1. Date despre program

Instituția de învățământ superior	Universitatea Lucian Blaga din Sibiu
Facultatea	Științe
Departament	Departamentul de Matematică și Informatică
Domeniul de studiu	Informatică
Ciclul de studii	Masterat
Specializarea	Sisteme și tehnologii Informatică avansate

### 2. Date despre disciplină

Denumirea disciplinei	Medii și platforme de dezvoltare avansate			
Codul cursului	Tipul cursului	An de studiu	Semestrul	Număr de credite
38031002006	DS	1	2	7
Tipul de evaluare	Categorია formativă a disciplinei (DA=aprofundare.; DS=sinteză)			
Examen	DA			
Titular activități curs	Lector univ. dr. Hunyadi Daniel			
Titular activități seminar / laborator/ proiect	Lector univ. dr. Hunyadi Daniel			

### 3. Timpul total estimat

Extinderea disciplinei în planul de învățământ – număr de ore pe săptămână				
Curs	Seminar	Laborator	Proiect	Total
2		2		4
Extinderea disciplinei în planul de învățământ – Total ore din planul de învățământ				
Curs	Seminar	Laborator	Proiect	Total (NOAD <sub>sem</sub> )
28		28		56

Distribuția fondului de timp pentru studiu individual		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		59
Tutoriat:		
Examinări:		
Total ore alocate studiului individual (NOSI <sub>sem</sub> )		119
Total ore pe semestru (NOAD <sub>sem</sub> + NOSI <sub>sem</sub> )		175

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

De curriculum	
De competențe	



# ULBS

Universitatea "Lucian Blaga" din Sibiu

Ministerul Educației și Cercetării

Universitatea "Lucian Blaga" din Sibiu

Prorector Programe Academice

## 5. Condiții (acolo unde este cazul)

De desfășurare a cursului	Sală de curs, dotată cu tablă, calculator și videoproiector
De desfășurare a sem/lab/pr	Sală de laborator dotată cu calculatoare desktop, software (Visual Studio, SQL Server), conexiune internet

## 6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"><li>Cunoașterea și utilizarea adecvată a noțiunilor teoretice fundamentale de programare orientată pe obiect</li><li>Cunoașterea și utilizarea conceptelor avansate de programare orientată pe obiect</li><li>Cunoașterea teoretică a principiilor robuste de programare</li><li>Capacitatea de a aplica principiile robuste în activitatea de programare</li><li>Cunoașterea și utilizarea adecvată a metodelor de refactoring pentru a face codul aplicațiilor mai ușor de menținut, mai rezistent și mai sigur</li><li>Cunoașterea teoretică și aplicarea practică a modelelor de design în programare</li><li>Capacitatea de a proiecta module complexe care aplică principiile robuste și modelele de design</li></ul>
Competențe transversale	<ul style="list-style-type: none"><li>Dezvoltarea atitudinii pozitive față de muncă și responsabilitate pentru propria pregătire profesională.</li><li>Dezvoltarea spiritului de muncă în echipă</li><li>Capacitatea de a pune accent pe rezultate și pe obținerea de produse funcționale</li></ul>

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

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

## 8. Conținuturi

Curs		Nr. ore
Curs 1	Recapitularea conceptelor de bază în programarea orientată pe obiect. Incapsulare, mostenire, polimorfism. Clase și instanțe. Static.	2
Curs 2	Recapitularea conceptelor avansate în programarea orientată pe obiect. Interfete și clase abstracte. Constructori. Mostenire versus compoziție.	2
Curs 3	Introducere în fundamentele tehnicilor de programare. Motivația și nevoile pe care le rezolvă. Prezentarea terminologiei folosită în dezvoltarea de software.	2
Curs 4	Introducerea conceptului de "clean code". "Refactoring" și "Software design". "Clean code" din punctul de vedere al experților din domeniu.	2
Curs 5	Tehnici de identificare a problemelor din cod. Identificarea greselilor: comentarii și denumirea variabilelor. Probleme generale.	2
Curs 6	Identificarea problemelor din cod: metode, clase, design	2

Curs 7	Denumirea robusta a variabilelor. Scrierea comentariilor. Metafora ziarului.	2
Curs 8	Standarde de codare. Standarde recunoscute in industrie	2
Curs 9	Modele de design. Modele creationale, structurale si comportamentale	2
Curs 10	Modele de design arhitecturale - MVC, MVVM. Repository and unit of work. Fluent interface.	2
Curs 11	Principii de programare. Principiile SOLID. Principiul responsabilitatii unice. Principiul deschis / inchis.	2
Curs 12	Principiul substitutiei Liskov. Principiul segregarii interfetelor. Principiul inversiunii de control.	2
Curs 13	Separarea intereselor in programare. Simplitatea in programare. Metodele DRY si YAGNI	2
Curs 14	Principii GRASP. Programare defensiva	2
<b>Total ore curs:</b>		<b>28</b>
<b>Seminar/Laborator</b>		Nr. ore
Sem 1	Incapsulare, moștenire, polimorfism	2
Sem 2	Source control. Exemple: Git & VSTS. Continuous integration	2
Sem 3	Exemple de cod. Refactoring	2
Sem 4	Alte exemple OOP. Mostenire vs compozitie	2
Sem 5	Code smells	2
Sem 6	Code smells	2
Sem 7	Coding guidelines	2
Sem 8	Coding guidelines	2
Sem 9	Design patterns	2
Sem 10	Design patterns	2
Sem 11	Design patterns	2
Sem 12	SOLID	2
Sem 13	Dependency injection	2
Sem 14	Coupling, cohesion, DRY	2
<b>Total ore seminar/laborator</b>		<b>28</b>

#### Metode de predare

Expunerea sistematică a cunoștințelor (deductivă, inductivă și formalizată, expuneri ppt); conversația frontală; conversația euristică, problematizare, studii de caz, modelarea, explicația, exemplificarea, învățarea prin descoperire

#### Bibliografie

Referințe bibliografice recomandate	<p>Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009</p> <p>Steve McConnell, Code Complete: A Practical Handbook of Software Construction, Microsoft Press, 2004</p> <p>Erich Gamma, Richard Helm, Ralph E. Johnson, John Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994</p>
-------------------------------------	---



# ULBS

Universitatea "Lucian Blaga" din Sibiu

Ministerul Educației și Cercetării

Universitatea "Lucian Blaga" din Sibiu

Prorector Programe Academice

Referințe bibliografice suplimentare	<a href="https://en.wikipedia.org/wiki/Design_Patterns">https://en.wikipedia.org/wiki/Design_Patterns</a> <a href="https://en.wikipedia.org/wiki/Software_design_pattern">https://en.wikipedia.org/wiki/Software_design_pattern</a> <a href="http://www.dofactory.com/net/design-patterns">http://www.dofactory.com/net/design-patterns</a> <a href="https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm">https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm</a>
--------------------------------------	--

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

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.

## 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Ponderea în nota finală	Obs.**
Curs	Înșușirea de cunoștințe fundamentale și aprofundate	Examen scris și aplicativ	50%	CEF
Laborator	Prezentare proiect	Evaluare pe parcurs	50%	nCPE

### Standard minim de performanță

Pentru promovarea examenului, candidatul trebuie să dețină cunoștințe fundamentale referitoare la programarea orientată obiect, să cunoască principii de programare de tipul clean code, code smells, design pattern. De asemenea studenții trebuie să realizeze în proporție de minim 50% proiectul de laborator.

(\*) Fișa disciplinei cuprinde componente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

(\*\*) CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală;

Data completării: 21.09.2020

Data avizării în Departament: 25.09.2020

	Grad didactic, titlul, prenume, numele	Semnătura
Titular disciplină	Lector univ. dr. Daniel Hunyadi	
Director de departament	Prof.univ.dr. Mugur Acu	