

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN ORADEA
1.2 Facultatea	FACULTATEA DE INGINERIE ELECTRICĂ ȘI TEHNOLOGIA INFORMAȚIEI
1.3 Departamentul	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.4 Domeniul de studii	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.5 Ciclul de studii	LICENȚĂ
1.6 Programul de studii/Calificarea	CALCULATOARE /INGINER

2. Date despre disciplină

2.1 Denumirea disciplinei	Proiectarea aplicațiilor orientate pe obiecte						
2.2 Titularul activităților de curs	Prof.univ.dr.ing. Zmaranda Doina						
2.3 Titularul activităților de laborator/proiect	Prof.univ.dr.ing. Zmaranda Doina						
2.4 Anul de studiu	IV	2.5 Semestrul	7	2.6 Tipul de evaluare	Ex	2.7 Regimul disciplinei	DS

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

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 laborator /proiect	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 laborator /proiect	28
Distribuția fondului de timp ore					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					2
Examinări					6
Alte activități.....					
3.7 Total ore studiu individual	44				
3.9 Total ore pe semestru	100				
3.10 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de competențe	Cunoașterea conceptelor de bază ale programării orientată pe obiecte

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	- cursul se desfășoară față în față (sală de curs dotată cu calculator și videoproiector); prezentare bazată pe slide-uri
5.2. de desfășurare a laboratorului /proiectului	- laboratorul se poate desfășura față în față (sală de laborator dotată cu calculatoare și platforma .NET /Visual Studio) - prezenta obligatorie la toate laboratoarele 100%; - recuperarea laboratoarelor se face conform Regulamentului privind activitatea profesională a studenților în baza Sistemului European de Credite Transferabile (ECTS); se pot recupera pe parcursul semestrului maxim 30% din totalul activităților

6.1. Competențele specifice acumulate	
Competențe profesionale	C1. Definește arhitectura software C3. Definește cerințe tehnice C4. Proiectează sistemul informatic C5. Remediază erorile din software
Competențe transversale	

6.2. Rezultatele așteptate ale învățării	
Cunoștințe	Studentul cunoaște și identifică concepte și metode specifice privitoare la utilizarea șabloanelor de proiectare obiectuală în conformitate cu principiile ingineriei software precum și cu modul acestora de aplicare în probleme concrete; studentul știe să utilizeze framework-uri și limbaje specifice pentru proiectare/modelare obiectuală (UML/Visual Paradigm) precum și un limbaj de programare orientat pe obiecte pentru implementarea în cadrul unor aplicații concrete (C#, Java)
Aptitudini	Studentul alege și utilizează concepte proprii proiectării/modelării obiectuale bazate pe șabloane de proiectare; specifică cerințele de proiectare, analizează, optimizează elaborează și dezvoltă o soluție scalabilă și extensibilă utilizând șabloane de proiectare specifice și aplicând principii de dezvoltare software
Responsabilitate și autonomie	Studentul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale; studentul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei

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

7.1 Obiectivul general al disciplinei	În contextul diversității aplicațiilor software existente, cu necesități stringente de performanță, cursul abordează aspectele specifice din domeniul proiectării și dezvoltării aplicațiilor orientate pe obiecte
7.2 Obiectivele specifice	- Cursul își propune prezentarea conceptelor teoretice și a șabloanelor care stau la baza proiectării aplicațiilor orientate pe obiecte - Laboratorul familiarizează studenții cu aspecte practice privind proiectarea, modelarea și implementarea de aplicații orientate pe obiecte utilizând șabloane de proiectare precum a uneltelor existente în domeniul analizei și proiectării obiectuale - limbajul UML (Unified Modeling Language). Implementările utilizează platforma .NET și limbajul C# , fără a restrânge însă generalitatea conceptelor prezentate

8. Conținuturi

8.1.Curs	Metode de predare	Observații
UML - Unified Modeling Language. Modelul și concept generale UML.	Expunere liberă, cu prezentarea cursului pe slide-uri, față în față; discuții libere;	2
Diagrame structurale și comportamentale în modelul UML.		2
Proiectarea orientată pe obiecte utilizând UML. Specificarea cerințelor. Analiza orientată pe obiecte: dezvoltarea diagramelor de clase de analiză		2
Proiectarea orientată pe obiecte utilizând UML. Dezvoltarea diagramelor use-case și a diagramelor sequence. Rafinarea modelului și realizarea diagramelor de clase de proiectare. Refactorizare.		2
Principiile de dezvoltare/proiectare SOLID. Sabloane de proiectare obiectuală – concepte. Clasificarea șabloanelor de proiectare. Aplicabilitate. Beneficii și avantaje.		2
Șabloane creaționale: Singleton, Factory, AbstractFactory, Builder, Prototype.		2
Exemple de șabloane creaționale.		2
Șabloane structurale: Façade, Decorator, Adapter, Bridge, Composite, Flyweight, Proxy.		2
Exemple de șabloane structurale.		2
Șabloane comportamentale: Visitor. State. Observer, Command, Strategy, Chain of Responsibility, Interpreter, Iterator, Mediator, Memento, Template.		4
Exemple de șabloane comportamentale.		4
Șabloane arhitecturale: MVC (Model-View-Controller). Repository		2
Bibliografie 1. D. Zmaranda - Proiectarea sistemelor orientate pe obiecte utilizând șabloane de proiectare, Editura Universității din Oradea,ISBN 978-606-10-0427-0, 332pg., 2011 2. D. Zaharie, D. Zmaranda - Dezvoltarea aplicațiilor software utilizând platforma .NET, Editura ASE București, ISBN 978-606-505-547-6, 506pg., 2012 3. Dathan, Brahma, Ramnath, Sarnath, Object-Oriented Analysis, Design and Implementation, An Integrated Approach, ISBN 978-3-319-24280-4, second edition, University Press, 2015 4. Gary Mclean, Adaptive Code via C#: Agile coding with design patterns and SOLID principles, ISBN-13 : 978-0735683204, Microsoft Press; 1st Edition, 2014 5. Dmitri Nesteruk, Design Patterns in .NET: Reusable Approaches in C# and F# for Object-Oriented Software Design, ISBN-13 : 978-1484243657, Apress; 1st ed. Edition, 2019 6. Jimmy Nilsson, Applying Domain-Driven Design and Patterns: With Examples in C# and .NET, Addison-Wesley, 2006 7. Martin Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language (3nd Edition), Addison Wesley – Pearson Education, 2004 8. Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) , Prentice Hall, 2004 9. https://www.oodesign.com/ 10. https://softwarepatternslexicon.com/object-oriented/1/1/ 11. https://www.geeksforgeeks.org/system-design/patterns-and-frameworks-in-ooad/ 12. https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm 13. https://uoradea-my.sharepoint.com/personal/rodica_zmaranda_didactic_uoradea_ro/_layouts/15/onedrive.aspx?isAscending=true&id=%2Fpersonal%2Frodica%5Fzmaranda%5Fdidactic%5Fuoradea%5Fro%2FDocuments%2FPAOO%2FCurs%5FPAOO&sortField=LinkFilename&view=0		

8.2. Laborator	Metode de predare	Observații
Conceptele de bază UML. Crearea de diagrame UML: diagrame structurale și diagrame comportamentale	Studentii primesc temele pentru laborator cu cel puțin o săptămână înainte, și le studiază. La începutul laboratorului se discută și se analizează modalitățile de rezolvare a aplicațiilor propuse, folosind videoproectorul. Pe urmă, studenții realizează partea practică a lucrării (problemele propuse) sub îndrumarea cadrului didactic.	2
Studiu de caz: aplicație pentru gestionarea proceselor dintr-o bibliotecă. Analiza orientată pe obiecte - discuții. Specificarea cerințelor/funcționalităților. Dezvoltarea modelului conceptual al sistemului		2
Studiu de caz: aplicație pentru gestionarea proceselor dintr-o bibliotecă. Proiectarea orientată pe obiecte: Identificarea subsistemelor majore ale aplicației; identificarea și crearea claselor software		4
Studiu de caz: aplicație pentru gestionarea proceselor dintr-o bibliotecă – implementare. Cuplare redusă. Utilizarea unui cod generic. Șablonul Façade. Șablonul Singleton		2
Studiu de caz: aplicație pentru gestionarea proceselor dintr-o bibliotecă. Extensibilitatea soluției. Șablonul Factory		2
Studiu de caz: aplicație pentru gestionarea proceselor dintr-o bibliotecă. Extensibilitatea soluției - refactorizare. Șablonul Decorator		2
Modelarea bazată pe Diagrame de Stări Finite (Finite State Machine FSM). Studiu de caz: controller pentru un cuptor cu microunde		2
Studiu de caz: controller pentru un cuptor cu microunde – refactorizare. Șablonul State		4
Studiu de caz: controller pentru un cuptor cu microunde – refactorizare. Șablonul Observer		2
Incheierea situației la laborator/verificări și recuperări		4
Bibliografie		
1. D. Zmaranda - Proiectarea sistemelor orientate pe obiecte utilizând șabloane de proiectare, Editura Universității din Oradea, ISBN 978-606-10-0427-0, 332pg., 2011		
2. https://uoradea-my.sharepoint.com/personal/rodica_zmaranda_didactic_uoradea_ro/_layouts/15/onedrive.aspx?isAscending=true&id=%2Fpersonal%2Frodica%5Fzmaranda%5Fdidactic%5Fuoradea%5Fro%2FDocuments%2FPAAO%2FLaborator%5FPAAO&sortField=LinkFilename&view=0		

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

În contextul creșterii impactului dezvoltării software, disciplina se regăsește în conținutul curriculei de la Universitatea Politehnica Timisoara. Cunoașterea conceptelor de bază ale modelării și proiectării orientate pe obiecte și a șabloanelor de proiectare, prezentate în cadrul acestei discipline, reprezintă o cerință importantă pentru dezvoltarea abilităților și abilităților de programare solicitate de companiile 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	- pentru nota 5 este necesară cunoașterea noțiunilor fundamentale solicitate în subiecte, fără a prezenta detalii asupra acestora, care să însumeze în total 5 puncte (4 puncte + 1 punct din oficiu) - pentru nota 10, este necesară cunoașterea amănunțită a tuturor subiectelor (10 puncte)	Examen scris - evaluarea se face față în față Studentii primesc spre rezolvare fiecare un subiect cu întrebări care testează însușirea conceptelor teoretice prezentate în cadrul cursului. Fiecare întrebare are un punctaj; în total întrebările însumează un punctaj de 9 de puncte; se acordă 1 punct din oficiu.	40 %

10.5 Laborator	- pentru nota 5, realizarea unui procent de minim 50% din aplicațiile propuse, explicarea modalității de implementare și a arhitecturii acestora - pentru nota 10, finalizarea corectă și completă tuturor aplicațiilor propuse	Aplicație practică - evaluarea se face față în față La fiecare laborator studenții sunt evaluați pe baza activității (răspunsuri la întrebări, propuneri de implementare, etc.), evaluări care se concretizează la finalul laboratorului într-o notă pe activitatea de laborator din cursul semestrului. De asemenea, până în ultima oră de laborator studenții finalizează și predau toate aplicațiile practice propuse în cadrul laboratorului. Media dintre nota primită pentru aplicațiile practice și nota de la activitatea de la laborator va reprezenta nota finală de laborator	60%
Calcul Nota disciplină: 0.60 Nota finală laborator + 0.40 Nota examen scris Conditii de participare la examenul final: Nota finală laborator ≥ 5 Conditii de promovare: Nota disciplină ≥ 5			
10.6 Standard minim de performanță			
- cunoașterea și înțelegerea conceptelor de bază specifice proiectării și implementării unei aplicații orientate pe obiecte precum și a uneltelor/limbajelor utilizate în domeniul modelării și proiectării obiectuale - cunoașterea și înțelegerea structurii generale a aplicațiilor obiectuale și familiarizarea cu șabloanele de proiectare specifice domeniului - cunoașterea aspectelor specifice proiectării și implementării unei aplicații orientate pe obiecte: conceptele fundamentale, structurarea aplicațiilor precum și a modalității de aplicare a conceptelor teoretice în cadrul dezvoltării unei aplicații practice, concrete - cunoașterea și utilizarea șabloanelor obiectuale și arhitecturale			

Data completării
02.09.2025

Semnătura titularului de curs,
Prof. dr. ing. Zmaranda Doina
dzmaranda@uoradea.ro

Semnătura titularului de laborator
Prof. dr. ing. Zmaranda Doina
dzmaranda@uoradea.ro

Data avizării în departament
24.09.2025

Semnătura directorului de departament,
Conf.univ. dr. Elisa Valentina Moisi
emoisi@uoradea.ro

Data avizării în consiliul facultății
25.09.2025

Semnătură Decan,
Conf. univ. dr. ing. Eugen Gergely
egergely@uoradea.ro