

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	Programare orientată pe obiecte						
2.2 Titularul activităților de curs	Prof.univ.dr.ing. Zmaranda Doina						
2.3 Titularul activităților de laborator/proiect	Drd. Artenie Andrada						
2.4 Anul de studiu	II	2.5 Semestrul	4	2.6 Tipul de evaluare	Ex	2.7 Regimul disciplinei	DD

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					12
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
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	Competențe de bază de programare

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 desfășoară 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 C5. Remediază erorile din software
Competențe transversale	

6.2. Rezultatele așteptate ale învățării	
Cunoștințe	Studentul cunoaște și descrie concepte și metode elementare privitoare la abordările și limbajele de programare orientate pe obiecte; studentul identifică medii de programare (framework-uri) și cunoaște modalitatea de utilizare a acestora pentru realizarea de implementări în cadrul unor aplicații concrete (Visual Studio și .NET)
Aptitudini	Studentul alege și utilizează concepte proprii specifice implementărilor bazate pe abordarea orientată pe obiecte; specifică cerințe, analizează, elaborează, dezvoltă și testează programe într-un limbaj de programare obiectual (C#)
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	Obiectivul disciplinei îl reprezintă dobândirea de cunoștințe și familiarizarea studenților cu modalitatea de a proiecta și implementa aplicații software utilizând tehnologia orientată pe obiecte.
7.2 Obiectivele specifice	▪ Pornind de la conceptele de bază ale programării orientate pe obiecte: încapsularea, moștenirea și polimorfismul, cursul devoltă și prezintă concepte avansate precum clase generice și abstracte, interfețe, colecții de obiecte, evenimente și delegați, atribute și mecanismul reflexiei, serializare și fire de execuție. Exemplificările sunt realizate în limbajul C#, fără a restrânge însă generalitatea conceptelor prezentate în acest sens. La finalul cursului sunt prezentate elemente de acces către bazele de date; utilizarea unui ORM. ▪ Laboratorul, realizat utilizând limbajul C# și utilizând platforma .NET /Visual Studio , familiarizează studenții cu aspecte practice privind rezolvarea diferitelor tipuri de probleme utilizând conceptele programării obiectuale

8. Conținuturi

8.1.Curs	Metode de predare	Observații
Paradigme de programare. Concepte de bază POO în C#: clase și obiecte; spații de nume.	Expunere liberă, cu prezentarea cursului pe slide-uri, față în față; discuții libere;	2
Încapsularea și controlul accesului.		2
Constructorii și destructorii. Tipuri referință și tipul valoare. Membri statici. Clase parțiale.		2
Moștenirea. Polimorfismul.		4
Clase abstracte. Clase generice.		2
Colecții de obiecte. Colecții .NET – colecții generice și colecții non-generice. Interogari pe colecții de obiecte utilizand Linq		2
Interfețe. Interfețe MicroSoft .NET		2
Serializarea binară și serializarea XML. Manipularea fișierelor XML		2
Evenimente și delegați; expresii Lambda		2
Atribute și mecanismul reflexiei		2
Fire de execuție		2
Accesul la baze de date în .NET; utilizarea unui ORM (Object Relational Mapper) - .NET Entity Framework. Maparea în Entity Framework; obiecte context		4

Bibliografie

1. Microsoft Developer Network, <http://msdn.microsoft.com>
2. <http://www.c-sharpcorner.com/>
3. Brian Gorman – Practical Entity Framework Core & Database Access for Enterprise Applications 2nd Edition, ISBN-13: 978-1-4842-7300-5, 797pg., 2022
4. Christian Nagel – C# and .NET 2021 Edition, Wiley & Sons, ISBN: 978-1-119-79720-3, 2021
5. Zaharie Dorin, **Zmaranda Doina** - Dezvoltarea aplicațiilor software utilizând platforma .NET, Editura ASE București, ISBN 978-606-505-547-6, 506pg., 2012
6. **D. Zmaranda** - Proiectarea sistemelor orientate pe obiecte utilizând șabloane de proiectare, Editura Universității, din Oradea, ISBN 978-606-10-0427-0, 332pg., 2011
7. **D. Zmaranda**, Elemente de programare orientată pe obiecte în limbajul C#, Editura Universității din Oradea, ISBN 978-973-759-522-5, 2008
8. **D. Zmaranda**, C. Rusu, M. Gligor, Programare orientată pe obiecte cu aplicații în Visual C++ , Editura Universității din Oradea, ISBN 973-613-681-7, 2004
9. Erich Gamma, Ralph Johnson, Richard Helm, Design Patterns = Sabloane de proiectare : elemente de software reutilizabil orientat pe obiect, Bucuresti: Teora, 2002
10. **D. Zmaranda**, Elemente de programare orientată pe obiecte utilizînd limbajul C++, Editura Universității din Oradea, ISBN 973-613-013-4, 2001
9. 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%2FPOO%2FPOO%5Fcurs&sortField=LinkFilename&view=0

8.2. Laborator	Metode de predare	Observații
Clase și obiecte în C#. Ierarhii de clase. Spații de nume	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 videoproiectorul. Pe urmă, studenții realizează partea practică a lucrării (problemele propuse) sub îndrumarea cadrului didactic.	2
Constructorii și destructorii. Clase abstracte		2
Moștenirea și ierarhia de clase. Supraîncărcarea metodelor/constructorilor		4
Polimorfismul și legarea dinamică		2
Colecții de obiecte. Colecții nongenerice .NET.		2
Clase generice și colecții generice .NET.		2
Interfețe		4
Serializarea		2
Evenimente și delegați. Programare bazată pe evenimente		2
Accesul la baze de date în .NET		2
Încheierea situației la laborator, verificări și recuperări		4

Bibliografie

1. **D.Zmaranda**, A. Nicula, Elemente de programare orientată pe obiecte în limbajul C# - îndrumător de laborator, Editura Universității din Oradea, ISBN 978-973-759-523-2, 2008 – versiune electronică actualizată 2022

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%2FPOO%2FLAB%5FPOO&sortField=LinkFilename&view=0

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

- Conținutul disciplinei se regăsește în curricula specializării de Calculatoare și din alte centre universitare care au acreditat aceste specializări (Universitatea „Politehnica” din Timișoara, Politehnica București) iar cunoașterea conceptelor de bază a programării obiectuale împreună cu modalitatea de aplicare a acestora în dezvoltarea unor aplicații concrete, prezentate în cadrul acestei discipline, reprezintă o cerință fundamentală în scopul formării deprinderilor și abilităților de programare necesare

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 sunt necesare răspunsuri corecte care să însumeze în total 5 puncte (4 puncte + 1 punct din oficiu) - pentru nota 10, sunt necesare răspunsuri corecte la toate întrebările din cadrul subiectului propus (10 puncte)	Examen scris - evaluarea 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 unei implementări funcționale în proporție de 50% a problemelor propuse la laborator - pentru nota 10, cunoașterea amănunțită a modalității de implementare practică a tuturor lucrărilor de laborator	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

10.7 Standard minim de performanță

- familiarizarea cu modalitatea de a dezvolta și implementa aplicații software utilizând tehnologia orientată pe obiecte
- înțelegerea conceptelor de bază și avansate în programarea orientată pe obiecte și anume: încapsulare, moștenire și polimorfism, împreună cu multitudinea elementelor și structurilor care stau la baza implementării acestor concepte: clase (inclusiv clase generice și clase abstracte), interfețe, funcții virtuale, mecanismul supraîncărcării funcțiilor și operatorilor
- familiarizarea cu alte concepte avansate precum serializarea obiectelor, utilizarea firelor de execuție, atribute, mecanismul reflexiei
- utilizarea platformei Microsoft .NET împreună cu Visual Studio pentru dezvoltarea de diferite tipuri de aplicații orientate pe obiecte în limbajul C#
- aplicarea principiului reutilizării codului prin folosirea diferitelor biblioteci de clase existente în implementarea aplicațiilor software orientate pe obiecte

Data completării
02.09.2025

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

Semnătura titularului de laborator
Drd. Artenie Andrada
andradaartenie@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