



## FIȘA DISCIPLINEI DEZVOLTAREA APLICATIILOR SOFTWARE

### 1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA „OVIDIUS” DIN CONSTANȚA
1.2 Facultatea	Matematică și Informatică
1.3 Departamentul	Matematică și Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclu de studii	Master
1.6 Programul de studii	Medii virtuale multi-modale distribuite
1.7 Anul universitar	2025-2026

### 2. Date despre disciplină

2.1 Denumirea disciplinei	Dezvoltarea sistemelor software						
2.2 Cod disciplină	MVMOD.1.1.01						
2.3 Titularul activităților de curs	Conf.dr. Puchianu Crenguța						
2.4 Titularul activităților aplicative	Conf.dr. Puchianu Crenguța						
2.5 Anul de studii	1	2.6 Semestrul	1	2.7 Tipul de evaluare	E	2.8 Regimul disciplinei */**	DF/DOB

\* DF – disciplină fundamentală, DS – disciplină de specializare, DC – disciplină complementară

\*\* DOB – disciplină obligatorie; DOP – disciplină opțională; DFA – Disciplină facultativă

### 3. Timpul total (ore pe semestru)

3.1 Număr de ore activități directe pe săptămână	4	din care: 3.2 curs	2	3.3 aplicații***	2
3.4 Total ore activități directe pe semestru	56	din care: 3.5 curs	28	3.6 aplicații	28
3.7 Total ore de studiu individual					69
Distribuția fondului de timp					[ore]
Studiul cărților, manualelor, suportului de curs, notițelor, bibliografie minimală recomandată					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					13
Pregătire seminar / laborator / proiect, teme, referate, portofolii și eseuri					12
Pregătire pentru prezentări sau verificări					10
Pregătire pentru examinarea finală					14
Alte activități: consultații					10
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

\*\*\* S - seminar; L - laborator; P - proiect

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

4.1 de curriculum	-
4.2 de rezultate ale învățării	-

### 5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)



5.1. de desfășurare a cursului	Sala de curs disponibilă
5.2. de desfășurare a laboratorului	Sala de laborator disponibilă

## 6. Obiectivele disciplinei

6.1 Obiectivul general al disciplinei	Descrierea adecvată a metodologiei orientată spre obiecte și a mecanismelor limbajului UML, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic. Modelarea unui sistem informațional. Dezvoltarea orientată spre obiecte a unui sistem software. Dezvoltarea unui proiect și elaborarea documentației aferente.
6.2 Obiectivele specifice	Executarea unor sarcini profesionale complexe, în condiții de autonomie și de independență profesională, implicând detectarea și rezolvarea problemelor conexe apărute în dezvoltarea sistemelor orientate spre obiecte. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar.

## 7. Rezultatele învățării

Cunoștințe	Studentul / Absolventul: <ul style="list-style-type: none"><li>- știe să aplice principiile și mecanismele metodologiei orientată spre obiecte pentru dezvoltarea unui sistem informațional</li><li>- modelează un sistem informațional</li><li>- cunoaște modelele necesare dezvoltării unui sistem software</li><li>- elaborează documentația aferentă dezvoltării unui sistem software</li></ul>
Aptitudini	Studentul / Absolventul: <ul style="list-style-type: none"><li>- aplică metodologia orientată spre obiecte pentru dezvoltarea sistemelor informaționale care rezolvă probleme reale de complexitate medie</li><li>- analizează și proiectează arhitectura unui sistem software</li><li>- folosește instrumente informatice specializate în realizarea diagramelor UML</li></ul>
Responsabilitate și autonomie	Studentul / Absolventul: <ul style="list-style-type: none"><li>- are o atitudine de discernământ în utilizarea surselor de informare</li><li>- propune mai multe soluții de dezvoltare a sistemelor informaționale</li><li>- realizează în echipă proiecte de dezvoltare a sistemelor software</li><li>- îndeplinește la termen, în mod riguros, sarcini profesionale cu grad mediu de complexitate</li></ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Număr ore alocate
Sisteme. Definiții. Tipuri de sisteme. Mecanisme de tratare a complexității sistemelor	Metode de predare-învățare interactive	1
Sisteme informaționale. Procesul de dezvoltare a sistemelor informaționale. Fazele procesului de dezvoltare		1



**UOC-PO-10 Anexa 3**

Modele ale sistemelor informationale. Modelul organizației. Modelul agenților business. Modelul business. Diagrama UML de activități. Diagrama UML de clase	Dialogul	5
Introducere in ingineria sistemelor software. Procesul software. Activitati de baza si auxiliare. Modele de proces software.	Problematizarea	1
Analiza cerințelor software. Identificarea cerintelor functionale si nefunctionale.	Conversatia	2
Modelul funcțional. Actori software. Identificarea si descrierea cazurilor de utilizare software. Diagrame de cazuri de utilizare software. Relații între cazuri de utilizare. Diagrama de secvențe de sistem. Diagrama de context	Metodele active și interactive cu multiple	4
Proiectarea orientata spre obiecte. Arhitecturi software. Descrieri arhitecturale. Puncte de vedere si vederi arhitecturale. Stilul arhitectural MVC	Metode care contribuie la dezvoltarea gândirii critice	2
Modelul comportamental. Diagrame UML de interacțiuni.	Interacțiunea,	2
Modele de proiectare. Definitie, descriere. Clasificarea modelelor de proiectare. Modele de atribuire a responsabilitatii obiectelor.	problematizarea, argumentarea	4
Diagrame UML de clase de proiectare. Construirea diagramelor de interacțiuni folosind modele de proiectare de atribuire a responsabilităților. Construirea diagramelor de clase de proiectare folosind modelul comportamental.	Sintetizarea/esențializarea informațiilor	6
	Învățarea independentă și prin cooperare	
	Generalizarea	
<b>Bibliografie</b>		
[1]. Rumbaugh J., Jacobson I., Booch G., The Unified Modeling Language. Reference Manual, Addison-Wesley, 1999		
[2]. Bruegge B., Dutoit A.H., Object-Oriented Software Engineering, Editura Prentice Hall, 2000		
[3]. Show M., Garlan D., Software Architecture. Perspectives on an Emerging Discipline, Prentice Hall, 1996		
[4]. Larman C., Applying UML and Patterns, Prentice Hall, 1998		
[5]. Maciaszeck L. A., Liong B. L., Practical Software Engineering. A Case Study Approach, Editura Addison-Wesley, 2005		
[6]. Martin J., Odel J.J., Object-Oriented Methods: a Foundation. UML Edition, Prentice Hall, 1998		
[7]. Pressman R.S. , Software Engineering: A Practitioner's Approach, 7/e McGraw-Hill, 2010		
[8]. Popovici D. M., Bogdan C., Rusu A., Chelai O., Nicola A., Medii virtuale multimodale distribuite, Editura Universitaria Craiova și Editura Prouniversitaria Bucuresti, 978- 606-26-0049-5, 2014, vol 1, 354 pag.		
[9]. Șerbănați L.D., Integrating Tools for Software Development, Yourdon Press Computing Series, Prentice Hall, 1992		
[10]. Cursuri si laboratoare încărcate pe Teams pe canalul asociat disciplinei		
<b>8.2 Aplicații (laborator)</b>	<b>Metode de predare</b>	<b>Număr ore alocate</b>
Modele ale sistemelor informationale. Modelul organizației. Modelul agenților business. Modelul business. Diagrama UML de activități. Diagrama UML de clase	Metode de invatare interactive	5 ore



**UOC-PO-10 Anexa 3**

<i>Introducere in ingineria sistemelor software. Procesul software. Activitati de baza si auxiliare. Modele de proces software.</i>	Dialogul	2 ore
<i>Analiza cerințelor software. Identificarea cerintelor functionale si nefunctionale.</i>		2 ore
<i>Modelul funcțional. Actori software. Identificarea si descrierea cazurilor de utilizare software. Diagrame de cazuri de utilizare software. Relații între cazuri de utilizare. Diagrama de secvente de sistem. Diagrama de context</i>	Problematicizarea	5 ore
<i>Proiectarea orientata spre obiecte. Arhitecturi software. Descrieri arhitecturale. Puncte de vedere si vederi arhitecturale. Stilul arhitectural MVC</i>	Metode active și interactive	2 ore
<i>Modelul comportamental. Diagrame UML de interacțiuni.</i>		2 ore
<i>Modele de proiectare. Definitie, descriere. Clasificarea modelelor de proiectare. Modele de atribuire a responsabilitatii obiectelor.</i>	Sintetizarea/esențializarea informațiilor	4 ore
<i>Diagrame UML de clase de proiectare. Construirea diagramelor de interacțiuni folosind modele de atribuire a responsabilităților. Construirea diagramelor de clase de proiectare folosind modelul comportamental.</i>	Generalizarea	6 ore
	Explicația	
<b>Bibliografie</b> [1]. R. Pressman, Software engineering, A Practitioner's Approach, McGraw Hill, 2010 [2]. Bruegge B., Dutoit A.H., Object-Oriented Software Engineering, Editura Prentice Hall, 2000 [3]. Fowler M., UML Distilled, Addison Wesley, 1997 [4]. Gamma E., Helm R., Johnson R., Vlissides J., Design Patterns. Elements of Reusable Object-Oriented Software, Addison Wesley, 1995 [5]. Larman C., Applying UML and Patterns, Prentice Hall, 1998 [6]. Maciaszeck L. A., Liong B. L., Practical Software Engineering. A Case Study Approach, Editura Addison-Wesley, 2005 [7]. Martin J., Odel J.J., Object-Oriented Methods: a Foundation. UML Edition, Prentice Hall, 1998 [8]. Cursuri si laboratoare încărcate pe Teams pe canalul asociat disciplinei		

**9. Evaluare**

Tip activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Participare activă la activități didactice	Evaluare orală	10%
9.5 Laborator	Completarea unui test grila	Evaluare scrisă/online	30%
	Realizarea si prezentarea unui proiect	Evaluare orală	50%
Din oficiu			10%
9.6 Standard minim de performanță / Condiții de promovare			
Realizarea modelelor din faza de analiză a cerintelor software.			



**UOC-PO-10 Anexa 3**

Data completării,

Titular activităților de curs,  
Conf. univ. dr. Crenguta Puchianu

Titular aplicații,  
Conf. univ. dr. Crenguta Puchianu

17.09.2025

Data avizării în Departament,

Director de Departament,  
Conf. univ. dr. Elena Pelican

24.09.2025

Decan,  
Conf. univ. dr. Aurelian Nicola