

FIȘA DISCIPLINEI
Inginerie Software
Anul universitar 2020-2021

1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Științe, Educație Fizică și Informatică
1.3	Departamentul	Matematică-Informatică
1.4	Domeniul de studii	Informatică
1.5	Ciclul de studii	Licență
1.6	Programul de studiu / calificarea	Informatică/ Informatică

2. Date despre disciplină

2.1	Denumirea disciplinei	Inginerie Software									
2.2	Titularul activităților de curs	Conf univ. dr. Doru Anastasiu Popescu									
2.3	Titularul activităților de seminar / laborator	Conf univ. dr. Doru Anastasiu Popescu									
2.4	Anul de studii	III	2.5	Semestrul	I	2.6	Tipul de evaluare	E	2.7	Regimul disciplinei	O

3. Timpul total estimat

3.1	Număr de ore pe săptămână	4	3.2	din care curs	2	3.3	S / L / P	2
3.4	Total ore din planul de învăț.	56	3.5	din care curs	20	3.6	S / L / P	20
Distribuția fondului de timp alocat studiului individual								ore
Studiu după manual, suport de curs, bibliografie și notițe								35
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								15
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								30
Tutorat								6
Examinări								8
Alte activități: Consultații								8
3.7	Total ore studiu individual			94				
3.8	Total ore pe semestru			150				
3.9	Număr de credite			5				

4. Precondiții (acolo unde este cazul)

4.1	De curriculum	Algoritmi și structuri de date. Proiectarea și implementarea algoritmilor. Sisteme de gestiune a bazelor de date. POO.
4.2	De competențe	Algoritmi, POO, Limbaje de programare

5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sala de curs dotată cu video-proiector
5.2	De desfășurare a laboratorului	Laborator cu calculatoare

6. Competențe specifice vizate

Competențe profesionale	<ul style="list-style-type: none"> Programarea în limbaje de nivel înalt. Dezvoltarea și întreținerea aplicațiilor informatice. Utilizarea instrumentelor informatice în context interdisciplinar. Utilizarea bazelor teoretice ale informaticii și a modelelor formale. Proiectarea și gestiunea bazelor de date.
Competențe transversale	<ul style="list-style-type: none"> Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatică de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională.

7. Obiectivele disciplinei

7.1	Obiectivul general al disciplinei	Obiectivul general al disciplinei consta în studiul și aplicarea de abordări sistematice, disciplinate și cuantificabile în dezvoltarea sistemelor software.
7.2	Obiectivele specifice	Obiective cognitive: Studiul și aplicarea proceselor de dezvoltare software Înțelegerea activităților specifice ingineriei software Cunoașterea metodelor ingineriei software Cunoașterea unor instrumente specifice ce asistă programatorul în procesul de specificare, proiectare și validare Cunoașterea unor metode de modelare și analiza performanța software

	<p>Obiective procedurale:</p> <ul style="list-style-type: none"> • Formarea abilităților de proiectare și implementare de algoritmi pentru diverse clase de probleme și aplicații. • Aplicarea unor algoritmi eficienți de căutare și sortare în rezolvarea de probleme și în realizarea de aplicații. • Formarea deprinderilor de investigare a problemelor din perspectiva diverselor metode posibile de rezolvare, în vederea obținerii unor soluții eficiente. • Dobândirea și dezvoltarea unui stil de programare profesionist. • Aplicarea proceselor, metodelor și instrumentelor studiate în proiecte software de dimensiuni mici și medii <p>Obiective atitudinale:</p> <p>Rigurozitate în proiectarea algoritmilor și în programare.</p>
--	---

8. Conținuturi

8.1. Curs		Nr. ore	Metode de predare	Observații Resurse folosite
1	Introducere. Problemele ingineriei software	2		
2	Ce este un model? Etapile dezvoltării unei aplicații software. Standardul UML. Tipuri de diagrame UML: Diagrama claselor, Diagramele de colaborare, diagramele secvențelor și diagramele de stare, Diagramele de (cazuri de) utilizare (Use Case), diagrama componentelor, diagrame de instalare (deployment), diagrama pachetelor (package). TOOL-uri pentru UML (ARGOUML, Microsoft Visio, UMLet, etc.)	4		
3	Metodologii de dezvoltare a sistemelor software Abordarea „codează și repară” Metodologia secvențială Metodologia cascadă Metodologia ciclică Metodologia spirală Metodologia hibridă ecluză Prototipizarea Modelul V Metodă de programare „agilă” Extreme Programming Metodologia Open Source Reverse Engineering Metodologia de dezvoltare Offshore	4	Explicația Descrierea și exemplificarea	
4	Proiecte. Ciclul de viață optim pentru derularea unui proiect Culegerea de specificații Analiza Proiectarea Implementarea și Testarea Validare și Integrare Calificare Punerea în funcțiune Mentineră Retragerea sau înlocuirea	4	Demonstrația Problematizarea Conversația euristică Exercițiul	Calculator Videoproiector
5	Medii de programare pentru: -Programare interactivă folosind blocuri (Scratch, Mindstorms) -Programare clasică și POO (C++, Java, Python) -Programare web (Javascript, PHP)	4		
5	Programare vizuală – componenta importantă în crearea aplicațiilor software (Java, C#, etc.). Mecanisme specifice. Testare și validare Mecanisme, scop. Verificarea și validarea statică și dinamică. Testare și depanare. Planificarea verificării. Structura unui plan de destare.	4		
6	Tipuri de testări. Structura unui sistem de testare. Testarea și validarea aplicațiilor web	4		
7	Prezentarea rezultatelor științifice sau aplicațiilor software.	2		

Modalitati de realizare a prezentarilor si articolelor stiintifice pentru workshop-uri, conferinte, simpozioane, jurnale.			
<i>Bibliografie</i>			
1. I. Sommerville. Software Engineering (6th, 7th, 8th, 9th editions). Addison Wesley (2001, 2004, 2006, 2010).			
2. T. Lethbridge, R. Laganier. Object-Oriented Software Engineering: Practical Software Development using UML and Java (2nd edition). McGraw-Hill, 2005. http://www.lloseng.com .			
3. E. Currie. The essence of Z. Prentice Hall, 1999.			
4. E.N. Todoran. Inginerie software: studii în prototipizare și specificare formală. Mediamira, ClujNapoca, 2006.			
5. Lenuta Alboae, Sabin Buraga, Servicii Web. Concepte de baza si implementari, Editura Polirom, 2006			
6. Sabin Buraga, Proiectarea site-urilor Web – ediția a doua, Polirom, Iasi, 2005			
7. Doru Anastasiu Popescu, Gabriel Boroghina, Web-Based Programming Model, 6th International Conference on Modeling, Simulation, and Applied Optimization, May 27-29, Istanbul, Turkey, IEEE Xplorer, pp. 1-4, 2015			
8. Doru Anastasiu Popescu, Bold Nicolae, Modeling and testing web applications by using a reduced number of web pages, International Journal of Engineering Research & Industrial Applications (IJERIA), Vol.7, No. IV, November, pp. 13-20, 2014			
9. Doru Anastasiu Popescu, Bazele Programării, Java după C++, Editura L&S Soft, 2019, ISBN: 978-973-88037-9-4, 2019			
10. Doru Anastasiu Popescu, Programarea robotilor lego folosind mediul Mindstorms EV3, RMI, vol., Nr. 1, 2018			
8.2. Aplicații: Seminar / Laborator / Teme de casă		Nr. ore	Observații Resurse folosite
1	Analiza unei probleme si etapele de rezolvare a ei. Exemple	2	Calculator Videoproiector
2	Rezolvarea unor probleme folosind modelarea obiect.	4	
3	Programare interactiva folosind blocuri (Scratch, Mindstorms)	4	
4	Rezolvarea unor probleme care folosesc medii de programare pentru Java si Python.	4	
5	Aplicatii care folosesc programare vizuala.	4	
6	Testarea programelor. Exemple.	4	
7	Tool-uri pentru testarea aplicatiilor web. Studiu comparativ.	2	
<i>Bibliografie</i>			
1. I. Sommerville. Software Engineering (6th, 7th, 8th, 9th editions). Addison Wesley (2001, 2004, 2006, 2010).			
2. T. Lethbridge, R. Laganier. Object-Oriented Software Engineering: Practical Software Development using UML and Java (2nd edition). McGraw-Hill, 2005. http://www.lloseng.com .			
3. E. Currie. The essence of Z. Prentice Hall, 1999.			
4. E.N. Todoran. Inginerie software: studii în prototipizare și specificare formală. Mediamira, ClujNapoca, 2006.			
5. Lenuta Alboae, Sabin Buraga, Servicii Web. Concepte de baza si implementari, Editura Polirom, 2006			
6. Sabin Buraga, Proiectarea site-urilor Web – ediția a doua, Polirom, Iasi, 2005			
7. Doru Anastasiu Popescu, Gabriel Boroghina, Web-Based Programming Model, 6th International Conference on Modeling, Simulation, and Applied Optimization, May 27-29, Istanbul, Turkey, IEEE Xplorer, pp. 1-4, 2015			
8. Doru Anastasiu Popescu, Bold Nicolae, Modeling and testing web applications by using a reduced number of web pages, International Journal of Engineering Research & Industrial Applications (IJERIA), Vol.7, No. IV, November, pp. 13-20, 2014			
9. Doru Anastasiu Popescu, Bazele Programării, Java după C++, Editura L&S Soft, 2019, ISBN: 978-973-88037-9-4, 2019			
10. Doru Anastasiu Popescu, Programarea robotilor lego folosind mediul Mindstorms EV3, RMI, vol., Nr. 1, 2018			

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

Competențele dobândite în cadrul disciplinei permit absolvenților să utilizeze eficient metodele și modelele din ingineria software în rezolvarea cerințelor aferente practicii și cercetării în domeniul informaticii.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Abilitati de rezolvare probleme	Examen scris	50
10.5 Seminar / Laborator / Tema de casă	Activitate (rezolvarea și implementarea problemelor propuse) Tema de casă	Activitate laborator Proiect Test	20% 20% 10%
10.6 Standard minim de performanță	*Set de cunoștințe minimale pentru promovarea examenului final: cunoașterea principiilor fundamentale ale inginerie software; implementări de bază folosind tehnologiile descrise.		

Data completării
19 septembrie 2020

Titular de curs,
Conf. univ. dr. Doru Anastasiu Popescu

Titular de seminar / laborator,
Conf. univ. dr. Doru Anastasiu Popescu

Data aprobării în Consiliul departamentului,
19 septembrie 2020

Director de departament,
(prestator)
Conf. univ. dr. Doru Constantin

Director de departament,
(beneficiar),
Conf. univ. dr. Doru Constantin