

FIȘA DISCIPLINEI

Ingineria programelor

1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Electronică, Comunicații și Calculatoare
1.3	Departamentul	Electronică, Calculatoare și Inginerie Electrică
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	Ingineria programelor									
2.2	Titularul activităților de curs	Lect. univ. dr. Cristina TUDOSE									
2.3	Titularul activităților de laborator	Dr. Adriana OPREA / Lect. univ. dr. Cristina TUDOSE									
2.4	Anul de studii	III	2.5	Semestrul	I	2.6	Tipul de evaluare	E	2.7	Regimul disciplinei	D/O

3. Timpul total estimat

3.1	Număr de ore pe săptămână	5	3.2	din care curs	2	3.3	laborator	2	3.4	proiect	1
3.5	Total ore din planul de învățământ	70	3.6	din care curs	28	3.7	laborator	28	3.8	proiect	14
Distribuția fondului de timp											ore
Studiul după manual, suport de curs, bibliografie și notițe											20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren											14
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri											8
Tutorat											2
Examinări											6
Alte activități											-
3.9	Total ore studiu individual	50									
3.10	Total ore pe semestru	120									
3.11	Număr de credite	5									

4. Precondiții (acolo unde este cazul)

4.1	De curriculum	Parcursirea cursurilor de <i>Programarea calculatoarelor și limbaje de programare și Programare orientată pe obiecte</i>
4.2	De competențe	Operarea cu fundamente științifice, ingineresti și ale informaticii

5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sală de curs cu o capacitate de minim 100 locuri dotată cu videoproiector și ecran de proiecție, 2 table.
5.2	De desfășurare a laboratorului	Sală de laborator de informatică, capacitate maximă 20 studenți/laborator.

6. Competențe specifice acumulate

Competențe profesionale	C6 Proiectarea sistemelor inteligente C6.1 Descrierea componentelor sistemelor inteligente C6.2 Utilizarea de instrumente specifice domeniului pentru explicarea funcționării sistemelor inteligente C6.3 Aplicarea principiilor și metodelor de bază pentru specificarea de soluții la probleme tipice utilizând sisteme inteligente C6.4 Alegerea criteriilor și metodelor de evaluare a calitatii, performanțelor și limitelor sistemelor inteligente C6.5 Dezvoltarea și implementarea de proiecte profesionale pentru sisteme inteligente
Competențe transversale	

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	- Familiarizarea cu conceptele de bază ale ingineriei software: procese software, cerințe, modelare, testare, dezvoltare.
7.2 Obiectivele specifice	<i>Obiective cognitive</i>

	<ul style="list-style-type: none"> - Cunoașterea etapelor realizării unui proiect software. <p><i>Obiective procedurale</i></p> <ul style="list-style-type: none"> - Formarea abilităților de modelare, testare și verificare a unui sistem. - Formarea deprinderii de a specifica și verifica formal modelul unui sistem. <p><i>Obiective atitudinale</i></p> <ul style="list-style-type: none"> - Rigurozitate în realizarea testării și verificării formale în programare.
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8. Conținuturi

8.1. Curs		Metode de predare	Observații Resurse folosite
1	Introducere în Ingineria Programelor <ul style="list-style-type: none"> 1.1. Scurt istoric al Ingineriei Programelor 1.2. Definiții generale ale Ingineriei Programelor 1.3. Concepte de bază ale Ingineriei Programelor <p style="text-align: right;">Timp alocat: 2 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică	Calculator, Videoproiector, Suport documentar
2	Procese software <ul style="list-style-type: none"> 2.1. Modele de procese software 2.2. Etape și riscuri 2.3. Definirea și managementul cerințe <p style="text-align: right;">Timp alocat: 4 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
3	UML (Unified Modelling Language) <ul style="list-style-type: none"> 3.1. Definirea modelului 3.2. Cazuri de utilizare, diagrame de clase, diagrame de obiecte, diagrame de interacțiune, diagrame de stări, diagrame de activitate 3.3. Studiu de caz <p style="text-align: right;">Timp alocat: 8 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
4	Metode de testare a programelor <ul style="list-style-type: none"> 4.1. Testare funcțională 4.2. Testare structurală 4.3. Alte metode de testare <p style="text-align: right;">Timp alocat: 6 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
5	Verificare și validare software <ul style="list-style-type: none"> 5.1. Generalități 5.2. Revizii, audituri, demonstrații formale 5.3. Metode de specificare formală <p style="text-align: right;">Timp alocat: 4 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
6	Managementul unui proiect; standarde de evaluare a calității <p style="text-align: right;">Timp alocat: 2 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
7	Modelarea, testarea și verificarea unor sisteme <ul style="list-style-type: none"> 7.1. Studiu de caz: sisteme P <p style="text-align: right;">Timp alocat: 2 ore</p>	Prelegerea Explicația Descrierea și exemplificarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar

Bibliografie:

1. I. Somerville, Software engineering, Ed. Addison Wesley, 1996 (2001).
2. F. Ipat, Modelare orientată pe obiecte cu UML, Ed. Universității Pitești, 2001.
3. M. Holcombe, F. Ipat, Correct Systems: Building a Business Process Solution, Ed. Springer Verlag, 1998.
4. G. Varvara, Ingineria Programării, Ed. ConsPress, 2013.
5. R. S. Pressman, Software Engineering : A practitioner's approach, Ed. McGraw Hill, 2001.
6. S. Schach, Classical and Object-Oriented Software Engineering: With UML and C++, Ed. McGraw Hill, 1999.
7. J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language reference manual, Ed. Addison Wesley, 2004.
8. F. Leon, Ingineria Programării - note de curs, format electronic, 2013.
9. F. Ostafi, Ingineria Sistemelor de programe - note de curs, format electronic, 2013.

10. J.C. Franchitti, Software Engineering, format electronic, New York University, 2009. 11. D. Roman, Ingineria programării obiectuale, Ed. Albastră, 1996. 12. M. Gheorghe, F. Ipate, On testing P systems, Membrane Computing – Ninth International Workshop, WMC 2008, Revised Selected and Invited Papers, Lecture Notes in Computer Science, vol. 5391, Springer, 2009, pp. 204–216. 13. Ghe. Păun, Membrane Computing: An Introduction, Springer, 2002. 14. M. Gheorghe, F. Ipate, C. Dragomir, A Kernel P System, Proceedings of 10th BWMC, 2012. 15. A. Țurcanu, Ingineria Programelor-note de curs, format electronic, 2014.			
8.2. Aplicații - Laborator		Metode de predare	Observații Resurse folosite
1	Elaborarea unor modele de procese software pornind de la scenariii date; managementul cerințelor Timp alocat: 4 ore	Exercițiul Explicația Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar
2	Modelarea folosind UML: cazuri de utilizare, diagrame de clase, diagrame de interacțiune, diagrame de stări, diagrame de activitate 2.1. Studii de caz pornind de la scenariii date 2.2. Utilizarea StarUML/ArgoUML pentru realizarea diagramelor 2.3. Generarea de cod folosind StarUML Timp alocat: 16 ore	Exercițiul Explicația Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar
3	Testarea și verificarea programelor 3.1. Exemplificarea unor metode de testare: formală, structurală, bazată pe mutații 3.2. Folosirea unor instrumente automate de testare: JUnit, MuJava, MuClipse Timp alocat: 4 ore	Exercițiul Explicația Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar
4	Managementul proiectelor și managementul calității 4.1. Studiu de caz 4.2. Utilizarea unor instrumente dedicate managementului proiectelor Timp alocat: 4 ore	Exercițiul Explicația Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar
Bibliografie: 1. F. Ipate, Modelare orientată pe obiecte cu UML, Ed. Universității Pitești, 2001. 2. J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language reference manual, Ed. Addison Wesley, 2004. 3. No Magic Inc., MagicDraw Tutorials, format electronic, 2011. 4. S. Schach, Classical and Object-Oriented Software Engineering: With UML and C++, Ed. McGraw Hill, 1999. 5. G. Varvara, Ingineria Programării, Ed. ConsPress, 2013. 6. J.C. Franchitti, Software Engineering, format electronic, New York University, 2009. 7. L. Vogel, JUnit Tutorial, http://www.vogella.com/tutorials/JUnit/article.html . 8. MuClipse Tutorial, http://muclipse.sourceforge.net . 9. Microsoft Project 2010 Tutorial, http://office.microsoft.com/en-us/project-help/getting-started-with-project-2010-RZ101831071.aspx?CTT=5&origin=HA104039046 . 10. A. Țurcanu, Ingineria Programelor-note de curs, format electronic, 2014.			
8.3. Aplicații - Proiect		Metode de predare	Observații Resurse folosite
1	Elaborarea unor modele de procese software pornind de la scenariii date; managementul cerințelor Timp alocat: 2 ore	Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar
2	Modelarea folosind UML: cazuri de utilizare, diagrame de clase, diagrame de interacțiune, diagrame de stări, diagrame de activitate Studii de caz. Dezvoltare proiect. Timp alocat: 8 ore	Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar
3	Testarea și verificarea programelor Studii de caz. Dezvoltare proiect. Timp alocat: 4 ore	Studiul de caz Dezbateră Lucrul în grup	Calculatoare, Tablă, Suport documentar

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 modelele și tehnicile specifice ingineriei programelor în rezolvarea cerințelor aferente practicii și cercetării în domeniul programării calculatoarelor și tehnologiei informației.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Evaluare finală	Probă scrisă	50%
10.5 Laborator	Activitate laborator Proiect	Verificare soluții Verificare proiect	20% 30%
10.6 Standard minim de performanță	<p>* Note de minim 5 la activitatea de laborator, la proiect și la evaluarea finală (rezolvarea în proporție de 50% a cerințelor); nota finală minim 5.</p> <p>* Set de cunoștințe minimale pentru promovarea examenului final:</p> <ul style="list-style-type: none"> - Cunoașterea principalelor noțiuni privind etapele elaborării unui proiect software; - Cunoașterea principalelor tehnici de modelare UML - Cunoașterea unor noțiuni fundamentale de testare și validare a programelor. 		

Data completării
20 septembrie 2018

Titular de curs
Lect. univ. dr. Cristina TUDOSE

Titular de laborator
Dr. Adriana OPREA,
Lect. univ. dr. Cristina TUDOSE

Data aprobării în Consiliul departamentului, Director de departament,
21 septembrie 2018 (prestator)

Conf. univ. dr. Doru CONSTANTIN

Director de departament,
(beneficiar)
Prof. univ. dr. ing. Gheorghe Șerban