FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Sapientia" din municipiul Cluj-Napoca
1.2. Facultatea/ DSPP	Facultatea de Științe Tehnice și Umaniste din Târgu Mureș
1.3. Domeniul de studii	Calculatoare și tehnologia informației
1.4. Ciclul de studii	Licență
1.5. Programul de studiu	Calculatoare
1.6. Calificarea	Inginer

2. Date despre disciplină

2.0. Departamentul				Departamentul de Inginerie Electrică			
2.1. Denumirea disciplinei				**Inginerie software**			
2.2. Titularul disciplinei / activităților de curs				Ș.l. dr. ing. Szántó Zoltán			
2.3. Titularul (ii) activităților de	seminar						
	laborator			Ș.l. dr. ing. Szántó Zoltán			
	proiect						
2.4. Anul de studiu	3	2.5. Semestrul	5	2.6. Tipul de evaluare	Ex.	2.7. Regimul disciplinei	DI

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. seminar / laborator/ proiect/ practică	2
3.4. Total ore din planul de învățământ	56	Din care: 3.5. curs	28	3.6. seminar/ laborator/ proiect/ practică	28

Distribuția fondului de timp:	ore
a) Studiul după manual, suport de curs, bibliografie și notițe	30
b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	12
c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	20
d) Tutoriat	5
e) Examinări	2
f) Alte activități:	

3.7. Total ore studiu individual	69
3.8. Total ore pe semestru	125
3.9. Numărul de puncte de credit	5

4. Precondiții (acolo unde este cazul)

4.1. De curriculum	Tehnici de programare, Inginerie software.
4.2. De competențe	Metodologii de proiectare, Structuri de date, Șabloane de proiectare de bază.

5. Condiții (acolo unde este cazul)

5.1. De desfășurare a cursului	Tablă, proiector, calculator.
5.2. De desfășurare a seminarului/laboratorului/proiectului	Calculatoare, software specific. Prezența la laborator este obligatorie.

6. Competențele specifice acumulate

Competențe profesionale	• C3 - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor • C3.1 - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice • C3.2 - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor • C3.3 - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor inginerești • C3.4 - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor • C3.5 - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete
Competențe transversale	CT2. Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate

7. Obiectivele disciplinei

7.1. Obiectivul general al disciplinei	Obiectivul general al disciplinei este prezentarea și analiza soluțiilor arhitecturale pe diferite nivele de abstractizare.
7.2. Obiectivele specifice	Pentru atingerea obiectivului principal se urmăresc obiectivele specifice: • Prezentarea și Analiza cerințelor funcționale și nefuncționale ale sistemelor și a tacticilor de rezolvare a acestora • Prezentarea și analiza stilurilor arhitecturale • Prezentarea și analiza șabloanelor arhitecturale și de proiectare • Prezentarea și analiza principiilor de proiectare a claselor și pachetelor

8. Conținuturi

8.1. Curs	Metode de predare	Observații
1. Introducere.	• Mijloace multimedia • Prezentări PowerPoint • Demonstrații pe tablă	2
2. Paradigme de dezvoltare a sistemelor software.		2
3. Modelarea cerințelor.		2
4. Limbajul UML. Proiectare OO.		2
5. Limbajul UML. Proiectare OO.		2
6. Principii de proiectare a claselor.		2
7. Principii de proiectare a pachetelor.		2
8. Aplicarea șabloanelor de proiectare (creaționale).		2
9. Aplicarea șabloanelor de proiectare (structurale).		2
10. Aplicarea șabloanelor de proiectare (comportamentale).		2
11. Principii SOLID.		2
12. Proiectare arhitecturală (Stiluri arhitecturale).		2
13. Proiectare arhitecturală (Stiluri arhitecturale).		2
14. Metrici ale calității proiectelor software.		2

Bibliografie obligatorie:

1. Software Engineering (10th Edition) – I. Sommerville, Pearson, 2015
2. Software Engineering: A Practitioner's Approach (8th Edition) – Roger Pressman, Bruce Maxim, McGraw-Hill Education, 2014

Bibliografie opțională

1. Case Study Research in Software Engineering - Guidelines and Examples – Per Runeson, Wiley, 2011
2. Software Engineering Reviews and Audits – Boyd L. Summers, Auerbach, 2011
3. Software Engineering. Architecture-Driven Software Development – R. Scmidt, M. Kaufmann, 2013
4. C# Design Pattern Essentials – Tony Bevis, Ability First Limited, 2012
5. Software Architecture in Practice (3rd Edition) – Len Bass, Paul Clements, Rick Kazman, Addison-Wesley Professional, 2012
6. Use Case Driven Object Modeling with UML: Theory and Practice (2nd Edition)– Doug Rosenberg, Matt Stephens, Apress, 2013
7. Object-Oriented Methods: A New Comprehensive Study (3rd Edition) – Richmond S. Adebiaye, CreateSpace Independent Publishing Platform, 2015
8. Soft Skills: The Software Developer's Life Manual – John Sonmez, Manning Publications, 2015
9. Practical Approach To Software Engineering: With UML Diagram – Sunny Singh, Satvik Vats, Anurag Vyas, LAP LAMBERT Academic Publishing, 2015

8.2. Seminar	Metode de predare	Observații

8.3. Laborator	Metode de predare	Observații
1. Obiecte și clase. Exerciții recapitulative de programare.	Implementarea temelor pe sistemele de calcul din laborator, prezentări și demonstrații la tablă	2
2. Diagrama cazurilor de utilizare.		2
3. Diagrama de stări. Diagrama de activități.		1
4. Diagrama de secvențe.		2
5. Diagrama de componente.		1
6. Diagrama de clase.		2
7. Șabloane de proiectare.		4
8. Fișa Cerințelor. Documentarea proiectelor.		4
9. Principii SOLID.		4
10. Testarea unităților.		4
11. Recuperare laboratoare/ Notare		2

Bibliografie obligatorie:

1. Software Engineering (10th Edition) – I. Sommerville, Pearson, 2015
2. Software Engineering: A Practitioner's Approach (8th Edition) – Roger Pressman, Bruce Maxim, McGraw-Hill Education, 2014

Bibliografie opțională

3. Case Study Research in Software Engineering - Guidelines and Examples – Per Runeson, Wiley, 2011
4. Software Engineering Reviews and Audits – Boyd L. Summers, Auerbach, 2011
5. Software Engineering. Architecture-Driven Software Development – R. Scmidt, M. Kaufmann, 2013
6. C# Design Pattern Essentials – Tony Bevis, Ability First Limited, 2012
7. Software Architecture in Practice (3rd Edition) – Len Bass, Paul Clements, Rick Kazman, Addison-Wesley Professional, 2012
8. Use Case Driven Object Modeling with UML: Theory and Practice (2nd Edition)– Doug Rosenberg, Matt Stephens, Apress, 2013
9. Object-Oriented Methods: A New Comprehensive Study (3rd Edition) – Richmond S. Adebiaye, CreateSpace Independent Publishing Platform, 2015
10. Soft Skills: The Software Developer's Life Manual – John Sonmez, Manning Publications, 2015
11. Practical Approach To Software Engineering: With UML Diagram – Sunny Singh, Satvik Vats,

Anurag Vyas, LAP LAMBERT Academic Publishing, 2015		
8.4. Proiect	Metode de predare	Observații
Nu este cazul		
8.5. Practica	Metode de predare	Observații
Nu este cazul		

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

Disciplina este o disciplină de domeniu în Calculatoare și Tehnologia Informației, conținutul ei fiind fundamental în proiectarea soluțiilor software. Conținutul disciplinei cuprinde atât soluțiile arhitecturale fundamentale cât și soluții moderne care adresează complexitatea actuală a sistemelor software. Conținutul este compatibil cu disciplinele similare predate la universități de prestigiu din țară și străinătate.

10. Evaluare

A. Condiții de îndeplinit pentru prezentarea la evaluare:

Pentru admiterea la examen este necesară nota de trecere la evaluarea de laborator, respectiv prezența la 70% dintre orele de curs. Pentru nota de trecere la laborator este necesară efectuarea tuturor lucrărilor de laborator,.

B. Criterii, metode și ponderi în evaluare:

Tip activitate		10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs		Cunoașterea diferitelor diagrame UML	Examen scris.	60%
10.5 .	Seminar			
	Laborator	Abilitați de rezolvare a problemelor, prezența, activitate.	Examen laborator.	40%
	Proiect		Proiect.	
10.6. Standard minim de performanță				
Dezvoltarea de soluții arhitecturale pentru specificațiile unor aplicații concrete.				

Semnătura titularului disciplinei:

Ș.l. dr. ing. Szántó Zoltán

Semnătura titularului /rilor de aplicații:

Ș.l. dr. ing. Szántó Zoltán

Data:

16. 09. 2019

Semnătura directorului de departament:

Conf. dr. ing. Domokos József

Semnătura coordonatorului programului de studii:

Ș.l.dr.ing. Szabó László Zsolt