

PROIECTAREA SISTEMELOR SOFTWARE
1. Date despre disciplină/modul

Facultatea	Calculatoare, Informatică și Microelectronică				
Departamentul	Inginerie Software și Automatică				
Ciclul de studii	Ciclul II, Studii superioare de master				
Programul de studii	Tehnologia informației				
Anul de studii	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
Anul I (învățământ cu frecvență)	I	PA	S – unite de curs de specialitate	O - unitate de curs obligatorie	5

2. Timpul total estimat

Total ore în planul de învățământ		Din care		
		Ore auditoriale	Lucrul individual	
		Proiect	Studiul materialului teoretic	Pregătire aplicații
Învățământ cu frecvență	150	40	40	70

3. Precondiții de acces la disciplină/modul

Conform planului de învățământ	Proiectare conceptuală a unei aplicații IT, Programarea calculatoarelor, Arhitectura calculatoarelor, Structuri de date și algoritmi, Tehnici și mecanisme de proiectare software.
Conform competențelor	Să demonstreze cunoașterea algoritmilor fundamentali, să folosească diverse modalități de comunicare în situații reale, să cunoască arhitectura calculatorului să demonstreze capacitate de aplicare a algoritmilor și a șabloanelor de proiectare.

4. Condiții de desfășurare a procesului educațional pentru

Proiect	- Pentru prezentarea materialului în sală este nevoie de tablă, tablă interactivă, de proiector și calculator. - Tematica proiectului trebuie să fie aleasă din tematicile disciplinei - Se lucrează individual - Acces la platforme software (IDE-uri, baze de date test) și resurse didactice. - Întâlniri săptămânale pentru raportarea progresului și primirea feedback-ului. - Documentația proiectului să includă specificații tehnice, diagrame de proiectare, prezentare și raport final. - Demonstrarea soluției și justificarea alegerilor în fața comisiei de evaluare.
----------------	--

5. Competențe specifice acumulate

Competențe profesionale	C1: Capacitatea de a analiza cerințele și de a defini arhitecturi software moderne: - Identificarea și clasificarea cerințelor funcționale și non-funcționale. - Alegerea și justificarea unei arhitecturi software corespunzătoare contextului proiectului. C2: Aplicarea șabloanelor de proiectare pentru crearea unui sistem scalabil: - Integrarea eficientă a șabloanelor de proiectare precum Singleton, Factory, Observer, etc. - Utilizarea corectă a principiilor de programare orientată pe obiecte și a bunelor practici de codare. C3: Dezvoltarea și testarea unui sistem software complet: - Implementarea corectă a soluțiilor propuse și verificarea funcționalității acestora. - Realizarea de teste unitare și integrarea lor în fluxul de dezvoltare.
Competențe transversale	CT1: Capacitatea de a lucra în echipe multidisciplinare: - Comunicare eficientă cu ceilalți membri ai echipei. - Alocarea și asumarea responsabilităților în funcție de abilități. CT2: Dezvoltarea spiritului analitic și a gândirii critice: - Evaluarea alternativelor tehnice pentru rezolvarea problemelor identificate.

	– Propunerea de soluții inovatoare și justificarea lor în fața unei audiențe. CT3: Respectarea principiilor etice și profesionale: – Asigurarea calității și integrității codului livrat. – Respectarea standardelor de documentare și raportare.
--	--

6. Obiectivele disciplinei/modulului

Obiectivul general	Pregătirea studenților pentru dezvoltarea unui sistem informatic utilizând arhitecturi moderne și șabloane de proiectare, respectând standardele de calitate.
Obiectivele specifice	Se dorește dezvoltarea următoarelor calități: • Familiarizarea cu arhitecturi software moderne. • Implementarea și integrarea șabloanelor de proiectare. • Crearea unei soluții scalabile și bine documentate.

7. Conținutul disciplinei/modulului

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
1. Introducere în arhitecturi software moderne	4
2. Șabloane de proiectare. Clasificare.	6
3. Șabloane de proiectare creaționale	6
4. Șabloane de proiectare structurale	6
5. Șabloane de proiectare comportamentale	6
6. Proiectarea sistemului: bune practici	4
7. Implementarea și testarea componentelor	4
8. Documentarea și prezentarea proiectului	4
Total:	40
Tematica lucrărilor practice	

8. Referințe bibliografice

Principale	1. Refactoring Guru, Design Patterns Tutorial. Available: https://refactoring.guru/design-patterns. 2. SourceMaking, Design Patterns Reference, Available: https://sourcemaking.com/design_patterns. 3. GeeksforGeeks, Design Patterns - Introduction, Available: https://www.geeksforgeeks.org/design-patterns-set-1-introduction/. 4. TutorialsPoint, Software Design Patterns, Available: https://www.tutorialspoint.com/design_pattern/index.htm. 5. Design Patterns Dev, Catalog of Design Patterns. Available: https://www.designpatterns.dev. 6. Java Design Patterns, Design Patterns Implemented in Java, Available: https://java-design-patterns.com.
Suplimentare	1. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, <i>Design Patterns: Elements of Reusable Object-Oriented Software</i>. Reading, MA, USA: Addison-Wesley, 1994. 2. R. C. Martin, <i>Clean Architecture: A Craftsman's Guide to Software Structure and Design</i>. Boston, MA, USA: Prentice Hall, 2017. 3. M. Grand, <i>Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML</i>, vol. 1-3. New York, NY, USA: Wiley, 2002. 4. C. Larman, <i>Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development</i>. Upper Saddle River, NJ, USA: Prentice Hall, 2004. 5. Shalloway and J. R. Trott, <i>Design Patterns Explained: A New Perspective on Object-Oriented Design</i>. Boston, MA, USA: Addison-Wesley, 2004.

9. Utilizarea IA generativă

Permisivitatea de utilizare	Utilizarea IA generative în cadrul temelor și proiectelor este permisă, cu condiția ca studenții să respecte următoarele reguli: - IA generativă poate fi utilizată pentru generarea de idei, structuri de text sau cod, dar toate materialele generate trebuie să fie revizuite și ajustate de către student pentru a se asigura că acestea corespund cerințelor academice. - Orice utilizare a IA generative trebuie să fie declarată în secțiunea de appendice a fiecărei lucrări, folosind fraza: "În timpul pregătirii acestei lucrări, autorul a utilizat [NUME INSTRUMENT / SERVICIU] în scopul [MOTIV]. După utilizarea acestui instrument/serviciu, autorul a revizuit și editat conținutul după cum a fost necesar și își asumă întreaga responsabilitate pentru conținutul lucrării."
Restricții de utilizare	Studenții nu trebuie să considere IA generativă ca o sursă de încredere pentru informații, deoarece nu oferă referințe clare sau surse documentate. - Nu este permisă citarea directă a conținutului generat de IA în lucrările academice ca și cum ar fi sursă primară. - Activitățile în care este interzis utilizarea IA generativă sunt specificare de profesor și sunt de regulă evaluări intermediare și finale sau care nu presupun activități de dezvoltare a competențelor profesionale.

10. Evaluare

Periodică și curentă		Evaluarea finală
Evaluare 1	Evaluare 2	
30%	30%	40%
Standard minim de performanță: definirea unei probleme a unui grup social și descrierea în ansamblu a soluției/soluțiilor utilizând tehnologia informației și comunicației.		
Prezența și activitatea la seminarele/atelierele de lucru; <i>Obținerea notei „5” la fiecare dintre evaluări;</i> <i>Obținerea notei „5” la lucrarea de examinare finală;</i> Evaluarea curentă, fiind de tip formativ și oferind studenților/echipei un feedback continuu la activitățile de proiectare sau modulele integrate, asigură evaluarea studentului cu nota echipei de lucru. Examenul final, fiind o evaluare sumativă, se realizează oral în baza proiectului prezentat public de echipă și discuții/interviuri individuale (în prezența echipei sau nu). Aprecierile obținute la examinare sunt individuale și constituie 40% din nota finală.		

11. Criterii de evaluare

Activitate	Componente evaluare	Metodă de evaluare, criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
Evaluare I	Prezența și participarea la seminare	Observație directă, fișă de prezență	50%	30 %
	Calitatea lucrărilor realizate în timpul atelierelor	Analiză practică, punctaj pentru fiecare sarcină rezolvată	50%	
Evaluare II	Prezentarea unei soluții funcționale	Evaluare practică, demonstrarea funcționalității soluției propuse	70%	30 %
	Colaborarea în echipă și respectarea termenelor	Feedback de la colegi și observațiile coordonatorului	30%	
Evaluarea finală	Prezentarea publică a proiectului final	Examinare orală, claritatea expunerii, justificarea deciziilor	60%	40%
	Documentația proiectului (conținut, structură, claritate)	Analiza documentației conform criteriilor predefinite	40%	