

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Transilvania din Brașov
1.2 Facultatea	Matematică și Informatică
1.3 Departamentul	Matematică și Informatică
1.4 Domeniul de studii de MASTERAT ¹⁾	Informatică
1.5 Ciclul de studii ²⁾	Masterat
1.6 Programul de studii/ Calificarea	Tehnologii moderne în ingineria sistemelor software

2. Date despre disciplină

2.1 Denumirea disciplinei	Dezvoltarea orientată pe componente							
2.2 Titularul activităților de curs								
2.3 Titularul activităților de seminar/ laborator/ proiect								
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Conținut ³⁾	DS
							Obligativitate ⁴⁾	DO

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	0/2/0
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/ laborator/ proiect	0/28/0
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					50
Tutoriat					14
Examinări					5
Alte activități.....					
3.7 Total ore de activitate a studentului					119
3.8 Total ore pe semestru	175				
3.9 Numărul de credite⁵⁾	7				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	- Concepte specifice pentru programarea orientată pe obiecte și dezvoltarea modulară. - Repere privind arhitectura și proiectarea aplicațiilor software și a sistemelor informatice. - Noțiuni fundamentale despre ingineria software. - Cunoștințe despre utilizarea unui sistem de control al versiunilor.
4.2 de competențe	- Abilitatea de a analiza și proiecta soluții software pentru probleme specifice. - Capacitatea de a documenta și prezenta soluții tehnice. - Etică profesională și rigoare în dezvoltarea software.

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	
5.2 de desfășurare a seminarului/ laboratorului/ proiectului	Sală de laborator cu resurse educaționale și informaționale specifice – calculatoare, conexiune de rețea, internet

6. Competențe specifice acumulate

Competențe profesionale	CP2. Stabilește procese de date, administrează sisteme de colectare a datelor, dezvoltă aplicații de procesare de date, implementează procese privind calitatea datelor, realizează extragerea informațiilor din date. - R.Î. 2.1. Absolventul utilizează instrumentele TIC pentru a aplica procese matematice, algoritmice sau alte procese de manipulare a datelor pentru a crea informații. - R.Î. 2.3. Absolventul creează software personalizat pentru prelucrarea datelor prin selectarea și utilizarea limbajului de programare informatică adecvat, pentru ca un sistem TIC să producă rezultate cerute pe baza contribuțiilor preconizate. CP5. Utilizează biblioteci de software, utilizează baze de date, creează softuri, proiectează interfețe între componente, utilizează șabloane de proiectare de software. - R.Î. 5.3. Absolventul transpune o serie de cerințe într-un concept de software clar și organizat. - R.Î. 5.5. Absolventul utilizează soluții reutilizabile, întocmește cele mai bune practici, în vederea îndeplinirii activităților comune de dezvoltare TIC în dezvoltarea și proiectarea de software.
Competențe transversale	CT3. Identifică probleme, soluționează probleme. - R.Î. 3.1. Absolventul identifică și detectează diverse probleme și aspecte și ia decizii cu privire la cea mai bună cale de urmat. - R.Î. 3.3. Absolventul găsește soluții la probleme practice, operaționale sau conceptuale într-o gamă largă de contexte. CT5. Lucrează în echipe, organizează informații, obiecte și resurse, dă dovadă de dorință de învățare, se adaptează la schimbare, face față stresului. - R.Î. 5.1. Absolventul lucrează cu încredere în cadrul unui grup, fiecare făcându-și partea lui în serviciul întregului. - R.Î. 5.3. Absolventul dă dovadă de o atitudine pozitivă față de cerințe noi și provocatoare care pot fi satisfăcute doar prin învățare pe tot parcursul vieții.

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	Familiarizarea studenților cu principiile, metodele și tehnologiile utilizate în dezvoltarea software orientată pe componente, precum și aplicarea lor practică în dezvoltarea aplicațiilor complexe.
7.2 Obiectivele specifice	- Înțelegerea arhitecturii orientate pe componente și a beneficiilor sale. - Utilizarea tehnologiilor și instrumentelor pentru dezvoltarea de componente software reutilizabile. - Implementarea și integrarea componentelor software în sisteme distribuite. - Dezvoltarea abilităților de proiectare și testare a aplicațiilor orientate pe componente.

8. Conținuturi

8.1 Curs	Metode de predare	Număr de ore	Observații
Introducere în dezvoltarea orientată pe componente. Principii de bază ale dezvoltării software pe componente. Caracteristici. Avantaje. Dezavantaje.	Curs interactiv	2	
Modelarea componentelor software. Modelarea componentelor cu diagrame UML.	Prelegere	4	
Design și implementare orientată pe componente. Șabloane de proiectare.		4	

Arhitecturi software bazate pe componente. Dezvoltarea orientată pe microservicii. Prezentarea arhitecturilor modulare bazate pe microservicii.	Dialog Dezbateri	8	
Tehnologii și framework-uri pentru dezvoltarea pe componente. Crearea și utilizarea serviciilor REST cu ASP.NET Core.		6	
Testarea și validarea componentelor. Testare unitară cu NUnit.Testarea serviciilor REST.		4	

8.2 Laborator	Metode de predare– învățare	Număr de ore	Observații
Crearea componentelor simple. Definirea și implementarea interfețelor. Utilizarea claselor și metodelor reutilizabile.		2	
Proiectarea și modelarea componentelor. Realizarea diagramelor UML pentru componente. Proiectarea interfețelor și a relațiilor dintre componente.		6	
Managementul dependențelor între componente.		2	
Dezvoltarea componentelor utilizând .NET. Dezvoltarea serviciilor REST utilizând ASP.NET Core.		6	
Testarea componentelor. Validarea serviciilor REST.		4	
Dezvoltarea microserviciilor în ASP.NET Core. Testarea și integrarea microserviciilor.		4	
Dezvoltarea unei aplicații complexe bazate pe componente și microservicii.		4	
Bibliografie			
1. Keith Short – “Component Based Development and Object Modeling” – Sterling Software, 1997 2. Alan W. Brown – “Building Systems from Pieces with Component-Based Software Engineering” - Chapter 6 of “Constructing Superior Software”, Macmillan Technical Publishing, 1999 3. Clemens Szyperski, Dominik Gruntz, Stephan Murer (2002). <i>Component Software: Beyond Object-Oriented Programming</i> . 2nd ed. ACM Press - Pearson Educational, London 2002 4. <i>Fundamentals of Software Architecture: An Engineering Approach</i> . O'Reilly Media. 2020. 5. Articole și resurse online (specificate pe parcursul semestrului).			

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

Conținuturile sunt în concordanță cu problematica monografiilor apărute în ultimii ani în literatura de specialitate.

10.Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4. Curs	Atingerea obiectivelor educaționale specifice disciplinei Acumularea competențelor vizate prin conținutul disciplinei	Evaluare finală prin probă scrisă	20%
10.5. Laborator		Evaluare formativă prin proiecte, dezvoltare de aplicații software, activități participative	80%
10.6. Standard minim de performanță			
Prezentarea la termene a activităților incluse în evaluarea formativă			

Prezenta Fișă de disciplină a fost avizată în ședința de Consiliu de departament din data de 26.09.2024 și aprobată în ședința de Consiliu al facultății din data de 26.09.2024.

Conf. dr. Gabriel Stan	Conf. dr. Nicușor Minculete
Decan	Director de departament
Titular de curs	Titular de laborator

Notă:

- ¹⁾ Domeniul de studii - se alege una din variantele: Licență/ Masterat/ Doctorat (se completează conform cu Nomenclatorul domeniilor și al specializărilor/ programelor de studii universitare în vigoare);
- ²⁾ Ciclul de studii - se alege una din variantele: Licență/ Masterat/ Doctorat;
- ³⁾ Regimul disciplinei (conținut) - se alege una din variantele: DF (disciplină fundamentală)/ DD (disciplină din domeniu)/ DS (disciplină de specialitate)/ DC (disciplină complementară) - pentru nivelul de licență; DAP (disciplină de aprofundare)/ DSI (disciplină de sinteză)/ DCA (disciplină de cunoaștere avansată) - pentru nivelul de masterat;
- ⁴⁾ Regimul disciplinei (obligativitate) - se alege una din variantele: DI (disciplină obligatorie)/ DO (disciplină opțională)/ DFac (disciplină facultativă);
- ⁵⁾ Un credit este echivalent cu 25 – 30 de ore de studiu (activități didactice și studiu individual).