

FIȘA DISCIPLINEI

Paradigme de Programare

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babes-Bolyai
1.2. Facultatea	Matematica si Informatica
1.3. Departamentul	Informatica
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Master
1.6. Programul de studii / Calificarea	Ingineria Software
1.7. Forma de învățământ	Cu frecventa

2. Date despre disciplină

2.1. Denumirea disciplinei		Paradigme de Programare				Codul disciplinei		MME8028			
2.2. Titularul activităților de curs			Conf.dr.ing. Florin Craciun								
2.3. Titularul activităților de seminar			Conf.dr.ing. Florin Craciun								
2.4. Anul de studiu		1	2.5. Semestrul		1	2.6. Tipul de evaluare		E	2.7. Regimul disciplinei		Obligativu

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	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					28
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					35
Tutoriat (consiliere profesională)					14
Examinări					14
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				119	
3.8. Total ore pe semestru				175	
3.9. Numărul de credite				7	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Fundamentele Programarii, Programarea orientata obiect, Metode Avnsate de programare, Programarea Logica si Functionala
4.2. de competențe	Competente medii de dezvoltare software

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	proiector
5.2. de desfășurare a seminarului/ laboratorului	proiector

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/ esențiale	• înțelegerea și operarea cu conceptele de bază din domeniul ingineriei software; • capacitatea de analiză și sinteză a informației; • modelare și rezolvarea de probleme din lumea reală; • însușirea conceptelor matematice și modelelor formale care să faciliteze înțelegerea, verificarea și validarea funcționării sistemelor software; • analiza, proiectarea și implementarea de sisteme software; • folosirea metodologiilor și instrumentelor specifice limbajelor de programare și ingineriei programării;
Competențe transversale	• abilități de muncă în echipă, cu preluarea diferitelor roluri de execuție și conducere pentru realizarea unor proiecte; • abilități de comunicare profesională: descrierea clară, concisă, verbală și în scris, a rezultatelor profesionale, abilități de negociere;

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: • Absolventul/a are cunoștințe necesare pentru a concepe, modela și proiecta aplicații software complexe • Absolventul/a cunoaște procese software și le poate integra în cultura organizațională a unei companii de software • Absolventul/a posedă cunoștințe fundamentale de modelare prin care analizează probleme din viața reală, le transpune în cerințe concrete și elaborează un model software corespunzător
Aptitudini	Studentul este capabil să • Absolventul/a demonstrează abilități avansate de programare care vor permite acumularea de cunoștințe solide și înțelegerea rapidă a tehnologiilor moderne din domeniu • Absolventul/a este în măsură să aplice cunoștințe avansate de sisteme informatice, plecând de la studierea la un nivel ridicat de abstractizare a diferitelor sisteme, fiind capabil/ă să ofere soluții de implementare pentru aplicații la sisteme informatice complexe, integrate • Absolventul/a este capabil să folosească limbajul de specialitate și terminologia specifică ingineriei software, astfel încât să poată comunica și interacționa cu membrii unor echipe de lucru
Responsabilități și autonomie	Studentul are capacitatea de a lucra independent pentru • Absolventul/a are capacitatea de a îmbina informațiile însușite în mod diferit prin combinarea acestora, în vederea formării unor atitudini pozitive în vederea dezvoltării proprii. • Absolventul/a este capabil/ă să realizeze cercetări în domeniul științelor educației, în special în domeniul gândirii algoritmice și gândirii critice • Absolventul/a are abilități de a realiza demersului de educare și pregătire pe diverse teme legate de dezvoltarea software • Absolventul/a are abilitatea de a analiza situațiile educaționale concrete în termenii unor norme și principii etice generale

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Cunoașterea și înțelegerea conceptelor fundamentale de programare. • Să fie capabil să aplice diferite paradigme de programare la diferite proiecte de programare
---------------------------------------	--

7.2 Obiectivele specifice	• să cunoască principalele caracteristici ale diferitelor paradigme de programare: procedurală, orientată pe obiecte, concurentă, funcțională, logică, bazată pe evenimente, scripting. • să aibă o bună înțelegere a următoarelor concepte: valoare, tip, variabilă, legătură, abstractizare procedurală, abstractizare de date, obiect, clasă, componentă, interfață, polimorfism; • să învețe asemănările și diferențele dintre diferitele paradigme de programare din punct de vedere al conceptelor pe care le implementează
----------------------------------	---

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Concepte de bază	• Expunere interactivă • Explicație • Conversație • Demonstrație interactivă	
2. Oz sintaxă, structuri de date		
3. Oz sintaxă, structuri de date		
4. Declarații, limbajul nucleului, mașină abstractă		
5. Programarea de ordin superior		
6. Calculul Lambda		
7. Recursivitate și excepții în tuplicat		
8. Tipuri, ADT, componente		
9. Concurența declarativă		
10. Concurența declarativă		
11. Concurența declarativă		
12. Programarea cu stare •		
13. Programarea relațională		
14. Programarea prin constrângeri		
Bibliografie 1. SCOTT, MICHAEL L.: Pragmatica limbajelor de programare, ed. a 4-a, Morgan-Kaufmann, 2016 2. SEBESTA, ROBERT W.: Concepts of Programming Languages, 10th ed, Pearson Education, 2012 3. SZYPERSKI, CLEMENS: Componente software. Beyond Object-Oriented Programming, Addison Wesley (ed. 1, 1998, ed. 2, 2002, cu GRUNTZ, DOMINIK și MURER, STEFAN). 4. STROUSTRUP, BJARNE: The C++ Programming Language Special Edition, Addison-Wesley, 2000 capitolul 2 5. VAN ROY, PETER; HARIDI, SEIF: Concepts, Techniques and Models of Computer Programming, MIT Press, 2004 6. WATT, David A.: Programming Language Design Concepts, Wiley, 2004		
8.2 Seminar / laborator	Metode de predare	Observații
Alocarea lucrărilor de cercetare pentru prezentarea orală	Utilizați instrumente practice pentru a implementa proiecte de grup. Discutați lucrări de cercetare.	Seminarul este organizat ca un total de 14 ore - 2 ore la fiecare două săptămâni Proiectul este organizat ca un total de 14 ore - 2 ore la fiecare două săptămâni.

Primul task de programare		
Prezentări de lucrări de cercetare		
Al doilea task de programare		
Prezentări de lucrări de cercetare		
Al treilea task de programare		
Prezentări de lucrări de cercetare		
Bibliografie Articole recente de cercetare Sistemul Mozart		

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

- Cursul respectă Recomandările IEEE și ACM Curriculare pentru studiile de Inginerie Software;
- Conținutul cursului este considerat de către companiile de software ca fiind important pentru abilitățile medii de dezvoltare software.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- să cunoască principiul de bază al domeniului; - să aplice conceptele cursului - rezolvarea problemelor	Examen final scris	40%
10.5 Seminar/laborator	- să fie capabili să aplice conceptele cursului - să fie capabili să facă o evaluare critică a lucrărilor de cercetare	-Prezentare de lucrări și aproximativ 3 taskuri de programare.	60%
10.6 Standard minim de performanță			
• Cel puțin nota 5 (pe o scară de la 1 la 10) atât la examenul final scris, cât și la lucrările de seminar.			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".

Nu se aplică.

Data completării:

...

Semnătura titularului de curs

Conf.dr.ing. Florin Craciun

Semnătura titularului de seminar

Conf.dr.ing. Florin Craciun

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA