

FIȘA DISCIPLINEI

Programare orientată obiect

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Facultatea de matematică și informatică
1.3. Departamentul	Departamentul de informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclu de studii	Licență
1.6. Programul de studii / Calificarea	Informatică în limba germană
1.7. Forma de învățământ	IF

2. Date despre disciplină

2.1. Denumirea disciplinei	Programare orientată obiect			Codul disciplinei	MLG5006		
2.2. Titularul activităților de curs	Ioan Crisan						
2.3. Titularul activităților de seminar	Ioan Crisan						
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligatorie (DS)

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/practic	1+2
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar	14+ 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)					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					20
Tutoriat (consiliere profesională)					14
Examinări					6
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				80	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Fundamentele Programării, Structuri de date și algoritmi
4.2. de competențe	Cunoștințe medii de programare într-un limbaj de programare de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	Sală de laborator cu calculatoare dotate cu limbajul de programare C++ și mediul de programare QT

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	<p>C3.1 Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice</p> <p>C3.2 Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor</p> <p>C3.4 Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor</p>
Competențe transversale	<p>CT1 Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei</p> <p>CT3 Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională</p>

6.2. Rezultatele învățării

Cunoștințe	<p>Studentul cunoaște:</p> <ul style="list-style-type: none"> • metodele, algoritmi, paradigme și tehnicile folosite în diferite ramuri ale informaticii. • utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor.
Aptitudini	<p>Studentul este capabil să :</p> <ul style="list-style-type: none"> • prezinte și să explice metodele, algoritmi, paradigme și tehnicile folosite în diferite ramuri ale informaticii. • folosească paradigme de programare (procedural, orientat obiect, funcțional) pentru realizarea de aplicații software adecvate specificului domeniului aplicat și dezvoltate. • înțeleagă și să comunice eficient informațiile.
Responsabilități și autonomie	<p>Studentul are capacitatea de a lucra independent pentru:</p> <ul style="list-style-type: none"> • a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu. • a înțelege și folosi conceptele programării orientate obiect la dezvoltarea unor aplicații software de complexitate medie. • conceperea programelor de calculator și analiza sistemelor software.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Să deprindă studentul cu proiectare orientată obiect a problemelor de scară mică/mijlocie și învățarea limbajului de programare C++ și a bibliotecii QT.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Demonstrarea diferenței dintre proiectarea imperativă tradițională și proiectarea orientată obiect. • Explicarea structurilor de tip clasă ca fiind componente fundamentale, modulare. • Înțelegerea rolului moștenirii, polimorfismului, legării dinamice și a structurilor generice în realizarea codului reutilizabil. • Explicarea și utilizarea strategiilor de programare DEFENSIVE, utilizarea aserțiunilor formale și tratarea excepțiilor. • Scrierea de programe de scară mică/mijlocie folosind C++ și QT. • Utilizarea claselor scrise de alți programatori în dezvoltarea sistemelor.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Paradigma de programare orientată obiect. <ul style="list-style-type: none">Elemente de bază ale limbajului C..Elemente lexicale. Operatori. Conversii.Tipuri de date. Variabile. Constante.Domeniul de vizibilitate și durata de viață a variabilelor. Spații de nume.Declararea și definirea funcțiilor. Supraîncărcarea funcțiilor. Funcții inline.	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaExempleDemonstrația didactică	
2. Programare modulară în C++. <ul style="list-style-type: none">Funcții. Parametrii.Fișiere header. Biblioteci.Implementarea modulară a TAD-urilor..Utilizarea pointerului void pentru obținerea genericității.	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaExempleDemonstrația didactică	
3. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++. <ul style="list-style-type: none">Tipuri de date: vectori și structuri.Tipuri de date: pointeri și referințe.Alocarea și dealocarea memoriei.Pointeri la funcții și pointeri spre void.	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaDemonstrația didactică	
4. Programare orientată obiect în C++. <ul style="list-style-type: none">Clase și obiecte.Membri unei clase. Modificatori de acces.Constructor/destructori.Diagrame UML pentru clase (membri, acces).	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaDemonstrația didactică	
5. Programare generică în C++	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaDemonstrația didactică	
6. Gestiunea resurselor și a memoriei în C++	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaDemonstrația didactică	
7. Moștenire <ul style="list-style-type: none">Moștenire simplă. Clase derivate.Principiul substituției.Supraîncărcarea metodelor.Moștenire multiplă.Relații de specializare/generalizare – reprezentări UML.	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaDemonstrația didactică	
8. Polimorfism	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversațiaDemonstrația didactică	
9. Interfață utilizator.	<ul style="list-style-type: none">Expunerea interactivăExplicațiaConversația	

	<ul style="list-style-type: none"> • Demonstrația didactică 	
10. Elemente de programare bazată pe evenimente I.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
11. Elemente de programare bazată pe evenimente II.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
12. Biblioteca STL <ul style="list-style-type: none"> • Iteratori STL. • Algoritmi STL. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
13. Aplicație POS (Point Of Sale) <ul style="list-style-type: none"> • Șabloane de proiectare Façade, Strategy. • Șablonul de proiectare Composite. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
14. Recapitulare	<ul style="list-style-type: none"> • Expunerea interactivă • Conversația 	

Bibliografie

- Bruce Eckel, Thinking in C++, www.bruceeckel.com
- Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002
- M. Frentiu, B. Parv, Elaborarea programelor. Metode si tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994.

In limba germana:

- G. Goos, W. Zimmermann, Objektorientiertes Programmieren und Algorithmen, Springer, Berlin, Heidelberg, New York, 2006.
- Pötzsch-Heffter, A., Konzepte objektorientierter Programmierung, Springer, Berlin, Heidelberg, 2009.
- Küchlin, W, Weber, A., Einführung in die Informatik, Objektorientiertes Programmieren mit Java, Springer, Berlin, Heidelberg, New York, 2004.
- B. Stroustup, Die C++ Programmiersprache, Addison Wesley, 2000.

8.2 Seminar	Metode de predare	Observații
		Seminarul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
1. Probleme simple în C. Funcții și parametric. Variabile locale și globale. Vizibilitate. Vectori (uni și multidimensionali) și structuri.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
2. TAD de tip container cu elemente generice (void*): reprezentări vizibile și ascunse.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
3. Clase. Clase simple. Supraîncărcarea operatorilor. Clase cu obiecte ca date membre.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
4. Clase de tip vector dinamic și iteratori. Moștenire.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația 	

	<ul style="list-style-type: none"> • Conversația • Demonstrația didactică • Exemple 	
5. Clase abstracte și interfețe. Polimorfism.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
6. Clase: template și excepții	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
7. Probleme complexe implementate pe baza diagramelor UML. Șabloane de proiectare. Pregătire pentru examenul scris.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică • Exemple 	
8.3 Laborator	<ul style="list-style-type: none"> • Metode de predare 	Observații
		<ul style="list-style-type: none"> • Laboratorul este structurat sub forma a 2 ore săptămânal. • Documentațiile de laborator se predau în săptămâna următoare celei în care s-a dat tema, iar programele se predau după două săptămâni.
1. Instalare Visual Studio. Specificare, proiectare și implementare probleme simple în C/C++. Aspecte generale C/C++.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
2. Programare modulară în C++	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
3. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversații • Modelarea 	
4. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
5. Proces de dezvoltare “feature driven”	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
6. Arhitecturi stratificate	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
7. Arhitecturi stratificate	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	

8. Arhitecturi stratificate	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
9. Fișiere text	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
10. GUI folosind QT	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
11. Repository.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
12. Containere, iteratori și algoritmi STL	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
13. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
14. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	

Bibliografie

- Bruce Eckel, Thinking in C++, www.bruceeckel.com
- Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002
- M. Frentiu, B. Parv, Elaborarea programelor. Metode si tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994.

In limba germana:

- G. Goos, W. Zimmermann, Objektorientiertes Programmieren und Algorithmen, Springer, Berlin, Heidelberg, New York, 2006.
- Pötzsch-Heffter, A., Konzepte objektorientierter Programmierung, Springer, Berlin, Heidelberg, 2009.
- Küchlin, W, Weber, A., Einführung in die Informatik, Objektorientiertes Programmieren mit Java, Springer, Berlin, Heidelberg, New York, 2004.
- B. Stroustup, Die C++ Programmiersprache, Addison Wesley, 2000.

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

- Conținutul disciplinei este în concordanță cu recomandările ACM pentru domeniul informatică.
- Cursul există în programul de studiu al universităților importante din țară și străinătate.
- Conținutul disciplinei este considerat de companiile software ca important pentru a asigura cunoștințe medii de programare orientată obiect.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	<ul style="list-style-type: none"> Corectitudinea și completitudinea cunoștințelor acumulate și capacitatea de a proiecta și implementa programe C++ 	Examen scris (în sesiunea regulată)	40%
10.5 Seminar/ Laborator	<ul style="list-style-type: none"> Capacitatea de a proiecta, testa și depana programe C++ folosind QT 	Evaluare practică (în sesiunea regulată)	30%
	<ul style="list-style-type: none"> Corectitudinea programelor C++ și a documentațiilor de laborator 	-documentații -portofoliu -observare continuă	30%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaștere și înțelegere a domeniului, că este capabil să exprime cunoștințele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni și de a utiliza cunoștințele în rezolvarea unor probleme în limbajul de programare C++. Pentru promovare este necesar ca nota finală să fie minim 5. 			

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

	Eticheta generală pentru Dezvoltare durabilă							

Data completării:
11.04.2025

Semnătura titularului de curs
Ioan Crișan

Semnătura titularului de seminar
Ioan Crișan

Data avizării în departament:
25.04.2025

Semnătura directorului de departament

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".

Conf. dr. Adrian Sterca