

FIȘA DISCIPLINEI

Programare pentru dispozitive mobile

Anul universitar 2025 - 2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Facultatea	Facultatea de Matematica 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	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare pentru dispozitive mobile / Mobile Application Programming / Programmierung für mobile Geräte				Codul disciplinei	MLG5078
2.2. Titularul activităților de curs			Lect. dr. Kuderna – Iulian BENȚA				
2.3. Titularul activităților de seminar			Lect. dr. Kuderna – Iulian BENȚA				
2.4. Anul de studiu	III	2.5. Semestrul	5	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligatorie

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

3.1. Număr de ore pe săptămână	3	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	1
3.4. Total ore din planul de învățământ	42	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	14
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					12
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					12
Tutoriat (consiliere profesională)					10
Examinări					12
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				58	
3.8. Total ore pe semestru				100	
3.9. Numărul de credite				4	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	• Algoritmă, structuri de date
4.2. de competențe	• Abilități medii de programare într-un limbaj de nivel înalt (orientat obiect), de preferință JAVA

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	• Videoproiector, acces la internet cu conexiune de mare viteză
5.2. de desfășurare a seminarului/ laboratorului	• Pentru activitatea de laborator este nevoie de calculatoare cu o viteză de procesare cât mai mare, memorie RAM minim 16 GB și conexiune de mare viteză la internet

6.1. Competențele specifice acumulate¹

Competențe profesionale/ esențiale	• programarea în limbaje de nivel înalt • dezvoltarea și întreținerea aplicațiilor informatice • utilizarea instrumentelor informatice în context interdisciplinar • utilizarea bazelor teoretice ale informaticii și a modelelor formale • proiectarea și gestiunea bazelor de date • proiectarea și administrarea rețelelor de calculatoare • folosirea conceptelor și tehnicilor inteligenței artificiale la rezolvarea unor probleme din lumea reală
Competențe transversale	• aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • desfășurarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea capacităților empatice de comunicare interpersonală, de relaționare și colaborare cu grupuri diverse • utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională.

6.2. Rezultatele învățării

Cunoștințe	• Studentul are cunoștințe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor. • Studentul are cunoștințe legate de programare, matematică, inginerie și tehnologie și are abilitățile necesare pentru a le folosi în crearea de sisteme informatice complexe. • Studentul are cunoștințele necesare pentru proiectarea, analiza și administrarea bazelor de date. • Studentul are cunoștințe adecvate legate de folosirea mediilor de dezvoltare integrate în scopul creării de aplicații complexe de dimensiuni mari. • Studentul cunoaște multiple limbaje de programare și este capabil să scrie aplicații în limbaje compilate, interpretate sau dinamice având capacitatea de a alege limbajul de programare potrivit pentru specificul aplicației de dezvoltat. • Studentul are cunoștințele necesare pentru selectarea și utilizarea procedurilor potrivite de instruire pentru a facilita procesul de asimilare a cunoștințelor. • Studentul are cunoștințe necesare pentru revizuirea literaturii de specialitate. • Studentul are cunoștințele necesare pentru procesarea și verificarea datelor și informațiilor. • Studentul are cunoștințele necesare legate de etapele ciclului de viață al softului și a modelelor de procese software. • Studentul cunoaște conceptele legate de modelarea softului și este capabil să implementeze cerințe funcționale și non-funcționale descrise în documente specifice pentru analiza și proiectarea sistemelor software. • Studentul are cunoștințele necesare pentru aplicarea tehnicilor de dezvoltare a softului pe baza modelelor. • Studentul are cunoștințele necesare legate de limbajul UML, precum și abilitatea de a utiliza instrumente CASE pentru a înțelege, documenta și implementa sisteme software. • Studentul cunoaște aspectele de bază legate de gestiunea softului. • Studentul este familiarizat cu metodologiile de dezvoltare tradiționale și agile. • Studentul cunoaște metodele de testare și verificare a sistemelor software. • Studentul are cunoștințe legate de bazele programării specifice sistemelor de operare, și are cunoștințe fundamentale în programarea în limbaje de tip script. • Studentul are cunoștințe fundamentale necesare instalării, configurării și întreținerii unui sistem server în Internet. • Studentul are cunoștințele adecvate legate de protocoale pe baza cărora funcționează rețeaua Internet, și are deprinderile necesare pentru proiectarea și testarea propriilor protocoale. • Studentul are cunoștințe necesare despre securitatea în internet și este capabil să aplice aceste cunoștințe pentru a valida și întreține o rețea de calculatoare care expune servicii uzuale, accesibile în mod securizat din exterior. • Studentul este familiar cu instrumente folosite pentru testarea, depanarea, validarea aplicației software. • Studentul este familiarizat cu instrumentele de management de proiect, sistemele de control al versiunilor, precum și conceptele, metodele, instrumentele de continuous integration/continuous delivery (CI/CD).
-------------------	---

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.

Aptitudini	• Studentul are aptitudinile necesare pentru conceperea programelor de calculator și analiza sistemelor software. • Studentul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații. • Studentul este capabil să aplice șabloane arhitecturale, șabloane de proiectare și bunele practici în domeniu pentru a proiecta aplicații software de complexitate mare. • Studentul are capacitatea de a evalua diferite arhitecturi și soluții posibile pentru o problemă și a alege pe cel potrivit pentru cerințele și constrângerile specifice aplicației de dezvoltat. • Studentul are aptitudinile necesare pentru înțelegerea și folosirea conceptelor programării orientate obiect la dezvoltarea unor aplicații software de complexitate medie. • Studentul are abilitatea de a înțelege și folosi șabloanele de proiectare pentru dezvoltarea aplicațiilor. • Studentul are deprinderile necesare pentru a dezvolta aplicații cu interfețe grafice folosind șabloane arhitecturale potrivite pentru specificul aplicațiilor cu interacțiune utilizator. • Studentul are abilitatea de a alege și folosi module și medii existente pentru dezvoltarea de aplicațiilor. • Studentul are abilitatea de a crea teste automate de diferite nivele de granularitate pentru asigurarea calității sistemelor dezvoltate. • Studentul este capabil de a realiza demersuri instructiv-educative, vizând dezvoltarea cognitivă a elevului, utilizând strategii și metode specifice educației în domeniul informaticii. • Studentul are abilitatea de a înțelege și comunica eficient informațiile. Studentul este capabil de a prezenta și a explica metodele, algoritmii, paradigmele și tehnicile folosite în diferite ramuri ale informaticii. • Studentul este capabil de a defini/identifica/înțelege probleme de cercetare în domeniul informaticii. • Studentul cunoaște bazele de date și bibliotecile digitale internaționale de cercetare academică (Web of Science, ACM Digital Library, IEEE Xplore, Springer, Elsevier, CiteSeerX, etc.). • Studentul are aptitudinile necesare pentru a aplica diferite metode și instrumente de vizualizare a rezultatelor cercetării. • Studentul are capacitatea de a observa și obține informații din diverse surse.
Responsabilități și autonomie	• Studentul are capacitatea de a alege și folosi paradigme de programare (procedural, orientat obiect, funcțional) pentru realizarea de aplicații software adecvate specificului domeniului aplicației dezvoltate. • Studentul are capacitatea de a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu. • Studentul are capacitatea de a identifica probleme complexe și de a examina probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții. • Studentul are capacitatea de a aplica reguli generale unor probleme specifice și de a produce soluții relevante. • Studentul are capacitatea de a utiliza instrumentele de sprijinire a cercetării.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Dezvoltarea abilității de a analiza, proiecta și implementa aplicații pentru dispozitive mobile
7.2 Obiectivele specifice	• Familiarizarea cu mediul de dezvoltare pentru aplicații mobile Android Studio • Abilitatea de a dezvolta aplicații simple cu accent pe interfața grafică, conectare cu alte dispozitive (internet, Bluetooth, NFC etc.), care preiau informații prin senzori (inclusiv de localizare), le stochează (local – în baza de date- sau distant) și le prelucrează.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere, sisteme de operare pentru dispozitive mobile și tehnologii	Expunerea interactivă Conversația Exemplificarea	2 ore
2. Introducere în Android – elemente de bază I	Expunerea interactivă Conversația Exemplificarea Problematizarea Demonstrația didactică	2 ore
3. Introducere în Android – elemente de bază II		2 ore
4. Interfața grafică (GUI) în Android		2 ore

5. Baza de date locală (SQLite)		2 ore
6. Aplicații distribuite		2 ore
7. Aplicații bazate pe localizare		2 ore
8. Aplicații bazate pe senzori I		2 ore
9. Aplicații bazate pe senzori II		
10. Aplicații adaptate la context		2 ore
11. Aplicații Cross-platform (React Native, Flutter)		2 ore
12. Introducere în iOS și Windows Phone		2 ore
13. Elemente avansate pentru realizarea aplicațiilor mobile complexe		2 ore
14. Prezentarea și discutarea proiectelor	Expunerea interactivă Conversația	2 ore

Bibliografie:

1. Hauke Fehr, "Eigene Apps programmieren: Schritt für Schritt mit LiveCode zur eigenen App – für Windows, Mac, iOS und Android", Editor Rheinwerk Verlag GmbH, 2016, ISBN-13: 978-3836243803
2. Vandad Nahavandipoor, „React Native Native Apps parallel für Android und iOS entwickeln“, ISBN Print: 978-3-96009-066-3
3. Jörg Staudemeyer, „Android mit Kotlin – kurz & gut: Inklusive Android 8 und Android Studio 3.0“, O'Reilly 2018, ISBN-13: 978-3960090380.
4. Becker, A., & Pant-Android, M. (2009). Grundlagen und Programmierung. dpunkt Verlag.
5. Becker, Arno, and Marcus Pant. "Android 2." Heidelberg: dpunkt (2010).
6. Thomas Theis, „Einstieg in Kotlin: Apps entwickeln mit Android Studio“, Rheinwerk Verlag, 2. aktualisierte Auflage, 2021.
7. Marc Marburger, „Flutter und Dart: Das umfassende Handbuch für die professionelle App-Entwicklung“, Rheinwerk Verlag, 1. Auflage, 2021.
8. Android Development. <http://developer.android.com/index.html>
9. Vogella. Android Development Tutorials. <http://www.vogella.com/android.html>
10. Michael Y. Morckos, Android Architecture, German University in Cairo, May 13, 2009
11. Reto Meier, Professional Android 2 Application Development, 2010
12. M. Cremene, I. K. Bența, "Dezvoltarea de aplicatii pentru terminale mobile", Ed. Albastra, Cluj-Napoca, 2006.

8.2 Seminar / laborator	Metode de predare	Observații
1. Prezentarea temelor de proiect, analiza și proiectarea aplicațiilor	Conversația Descoperirea Studiul individual Brainstorming-ul	2 ore
2. Aplicații simple în Android Studio, proiectarea și realizarea interfeței grafice	Conversația Problematizarea Studiul de caz Cooperarea Studiul individual Exercițiul	2 ore
3. Proiectarea și implementarea bazei de date local	Conversația Algoritmizarea Problematizarea Descoperirea Cooperarea Studiul individual Exercițiul	2 ore
4. Aplicații distribuite (TCP, HTTP, SOAP, JSON)	Conversația Algoritmizarea Problematizarea	2 ore
5. Aplicații bazate pe localizare și senzori	Studiul de caz Brainstorming-ul Studiul individual Exercițiul	2 ore
6. Dezvoltarea aplicațiilor de la proiect	Conversația Algoritmizarea	2 ore

	Studiul de caz Simularea Cooperarea Studiul individual	
7. Rafinarea și finalizarea aplicațiilor de la proiect	Conversația Algoritmizarea Problematizarea Studiul de caz	2 ore

Bibliografie:

1. Hauke Fehr, "Eigene Apps programmieren: Schritt für Schritt mit LiveCode zur eigenen App – für Windows, Mac, iOS und Android", Editor Rheinwerk Verlag GmbH, 2016, ISBN-13: 978-3836243803
2. Vandad Nahavandipoor, „React Native Native Apps parallel für Android und iOS entwickeln“, ISBN Print: 978-3-96009-066-3
3. Jörg Staudemeyer, „Android mit Kotlin – kurz & gut: Inklusive Android 8 und Android Studio 3.0“, O'Reilly 2018, ISBN-13: 978-3960090380.
4. Becker, A., & Pant-Android, M. (2009). Grundlagen und Programmierung. dpunkt Verlag.
5. Becker, Arno, and Marcus Pant. "Android 2." Heidelberg: dpunkt (2010).
6. Thomas Theis, „Einstieg in Kotlin: Apps entwickeln mit Android Studio“, Rheinwerk Verlag, 2. aktualisierte Auflage, 2021.
7. Marc Marburger, „Flutter und Dart: Das umfassende Handbuch für die professionelle App-Entwicklung“, Rheinwerk Verlag, 1. Auflage, 2021.
8. S. Android Development. <http://developer.android.com/index.html>
9. Vogella. Android Development Tutorials. <http://www.vogella.com/android.html>
10. Michael Y. Morckos, Android Architecture, German University in Cairo, May 13, 2009
11. Reto Meier, Professional Android 2 Application Development, 2010
12. M. Cremene, I. K. Bența, „Dezvoltarea de aplicatii pentru terminale mobile“, Ed. Albastra, Cluj-Napoca, 2006.

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

- Cursul respectă recomandările curriculare IEEE și ACM pentru studiile în informatică
- Cursul există în programa de studiu a majorității facultăților de profil din România
- Cursul există în programa de studiu a numeroase facultăți de profil din întreaga lume
- Companiile de software consideră conținutul cursului ca fiind util în dezvoltarea abilităților de modelare și programare ale studenților

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- Cunoașterea conceptelor de bază ale domeniului - Aplicarea principiilor OOP și tehnologiilor specifice aplicațiilor mobile pentru dezvoltarea de aplicații reale	Examen scris	30%
10.5 Seminar/ laborator	- Analiza, proiectarea, implementarea și testarea aplicațiilor mobile de la proiect	Prezentare proiect	50%
	- Analiza, proiectarea, implementarea și testarea aplicațiilor mobile de la laborator	Observarea sistematică a studentului în timpul rezolvării sarcinilor de la laborator.	20%
10.6 Standard minim de performanță			
• 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 proiectarea și implementarea de aplicații care rezolvă probleme reale.			

- Pentru a promova examenul la PDM trebuie să:
 - fie realizată o aplicație Android originală, minimală cu interfață grafică și încă o facilitare (folosirea bazei de date, a senzorilor, a GPS etc.)
 - media evaluării (examen scris, laborator) să fie peste 5.

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

Nu se aplică.

Data completării:
Aprilie 2025

Semnătura titularului de curs
Lector Dr. ing. Kuderna-Iulian Bența

Semnătura titularului de seminar
Lector Dr. ing. Kuderna-Iulian Bența

Data avizării în departament:
Aprilie 2025

Semnătura directorului de departament
Conf. dr. Adrian STERCA

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".