

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Tibiscus” din Timișoara
1.2. Facultatea	Calculatoare și Informatică Aplicată
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Master
1.6. Programul de studii/Calificarea	Administrarea Sistemelor Distribuite

2. Date despre disciplină

2.1. Denumirea disciplinei	Informatică aplicată (IA) opțional P1 – MAS115						
2.2. Titularul activității de curs	Conf.univ.dr. Florentina-Anica PINTEA						
2.3. Titularul activității de laborator	Conf.univ.dr. Florentina-Anica PINTEA						
2.4. Anul de studiu	1	2.5. Semestrul	1	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DA=aprofundare DOP=opțională

3. Timpul total estimat

3.1. Numărul de ore pe săptămână	4	din care 3.2. curs	2	3.3. laborator	2
3.4. Total ore din planul de învățământ	56	din care 3.5. curs	28	3.6. laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					28
Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate					18
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					10
Examinări					4
Alte activități					6
3.7. Total ore studiu individual	94				
3.8. Total ore pe semestru	150				
3.9. Numărul de credite	6				

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Tehnici de programare, Programare orientată obiect, Structuri de date și algoritmi, Baze de date și sisteme de gestiune a bazelor de date, Sisteme de operare, Rețele de calculatoare, Arhitectura sistemelor distribuite, Elemente fundamentale de securitate informatică
4.2. de competențe	Cunoașterea și înțelegerea modului de programare, Utilizarea mediilor de dezvoltare software și a instrumentelor de depanare, Înțelegerea principiilor de funcționare a sistemelor distribuite, Abilități de documentare tehnică și utilizare a literaturii de specialitate

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs dotată cu videoproiector, ecran de proiecție și acces la internet Calculator/laptop pentru cadrul didactic Acces la platformă e-learning (Google Classroom) pentru distribuirea materialelor și comunicare Software specific domeniului Posibilitatea susținerii cursului în format hibrid/online, dacă este cazul
5.2. de desfășurare a laboratorului	Laborator de informatică dotat cu calculatoare conectate în rețea și acces la internet Medii de dezvoltare necesare (ex. Python/Java, IDE-uri specifice, instrumente de administrare sisteme distribuite) Platformă de versionare (ex. Git) pentru gestionarea proiectelor

6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

Cunoștințe	Dezvoltă aplicații distribuite, scrie cod, implementează funcționalități Proiectează sisteme complexe, arhitecturi distribuite și soluții scalabile; Analizează performanțe, optimizează arhitecturi software; Optimizează algoritmi și arhitectura aplicației; Gestionează baze de date distribuite, replicare, securitate; Implementează aplicații de prelucrare a datelor;
Abilități	La finalizarea disciplinei, studentul masterand este capabil să: Dezvolte aplicații distribuite și să implementeze funcționalități complexe prin scrierea de cod adecvat; Proiecteze sisteme software distribuite și arhitecturi scalabile; Analizeze performanța aplicațiilor și să optimizeze algoritmi și arhitectura software; Gestioneze baze de date distribuite, implementând mecanisme de replicare și securitate;
Responsabilitate și autonomie	Își asumă responsabilitatea pentru proiectarea și implementarea soluțiilor distribuite; Demonstrează autonomie în analiza și rezolvarea problemelor complexe din domeniu; Adoptă bune practici privind securitatea, performanța și calitatea codului;

7. Conținuturi

7.1. Curs	Metode de predare	Observații
1. Elemente introductive, Clase și obiecte în Java Excepții și manipularea acestora	Expunere interactivă, problematizare, conversație euristică, documentare web, exemplificare	Dezvoltarea capacității de modelare a obiectelor și gestionare a fluxului de execuție. Formarea competențelor privind scrierea de cod robust și gestionarea excepțiilor.
2. Mini-aplicații (Applets)	Expunere, demonstrație practică	Exemplificare aplicații grafice
3. Interfețe grafice în Java (AWT, Swing)	Expunere interactivă, studiu de caz	Dezvoltarea aplicațiilor desktop
4. Fire de execuție (Thread) în Java	Expunere, modelare, exemplificare	Concurență și paralelism
5. Java Database Connectivity- JDBC	Expunere, demonstrație practică	Conectare la baze de date
6. Java Servlets	Expunere, studiu de caz	Aplicații web dinamice
7. Java 2 Micro Edition (J2ME)	Expunere, analiză comparativă	Aplicații pentru dispozitive mobile
8. Introducere în RPA - Robotic Process Automation	Expunere interactivă, documentare web	Corelare cu automatizarea proceselor
9. Mediu de configurare și cerințe preliminare în UiPath Studio	Demonstrație practică	Instalare și configurare
10. Caracteristici UiPath Studio	Expunere, exemplificare	Componente și arhitectură
11. Citirea și scrierea datelor în UiPath	Demonstrație practică	Automatizare documente și fișiere
12. 13. 14. Dezvoltarea unui proiect de automatizare folosind software-ul gratuit UiPath Studio (Academic Alliance Edition) pentru a automatiza anumite soluții	Învățare prin proiect	Implementare soluție completă
Resursele necesare învățării (cursuri, prezentări, materiale suport) sunt discutate la curs și disponibile pe Google Classroom		
7.2. Laborator	Metode de predare/învățare	Observații
1. Prezentarea mediului de dezvoltare Java (SDK), configurare proiecte	Demonstrație practică, exercițiu individual	Configurare IDE, structură proiect
2. Elemente de bază ale limbajului Java. Instrucțiuni și structuri de control	Exercițiul, problematizare	Aplicații simple
3. Clase și obiecte în Java. Constructori. Metode statice	Modelare, lucru în grup organizat	Crearea și utilizarea obiectelor
4. Moștenire, suprascriere metode, încapsulare, clase abstracte	Exercițiu aplicativ, dezbateri	Principii OOP
5. Tratarea excepțiilor în Java. Aplicații practice	Exercițiu, proiect aplicativ	Gestionarea erorilor

6. Interfețe grafice: AWT și Swing (JFrame, JPanel, JButton etc.)	Demonstrație practică, proiect	Dezvoltare aplicații GUI
7. Fire de execuție (Threads) în Java	Exercițiu aplicativ, modelare	Concurență
8. JDBC – conectare și operare cu baze de date	Exercițiu practic, proiect	CRUD aplicații
9. Java Servlets – aplicații web dinamice	Demonstrație practică, studiu de caz	Integrare web
10. Introducere în RPA și mediul UiPath Studio	Demonstrație practică	Instalare și configurare
11. Interfața UiPath Studio. Activități de bază	Exercițiu aplicativ	Primul flux de automatizare
12. Automatizarea citirii și scrierii datelor	Proiect aplicativ, lucru în grup	Fișiere Excel, documente
13. Automatizarea proceselor web și integrarea extensiilor browser	Exercițiu, modelare	Automatizare browser
14. Proiect final – dezvoltarea unei soluții complete de automatizare (Java sau RPA)	Învățare prin proiect (Project-Based Learning), prezentare și susținere	Evaluare integrată (funcționalitate, documentație, argumentare)

Materialele didactice și resursele necesare vor fi disponibile pe platforma Google Classroom.

Bibliografie

Pentru Java

1. Thinking in Java – Bruce Eckel, Prentice Hall, 2006.
2. Effective Java – Joshua Bloch, Addison-Wesley, 2018.
3. Core Java Volume I – Fundamentals – Cay S. Horstmann, Pearson.
4. Object-Oriented Analysis and Design – Grady Booch, Addison-Wesley.
5. Internetworking with TCP/IP Volume III – Douglas Comer, David Stevens, Prentice Hall.
6. Oracle – Documentație oficială Java: <https://docs.oracle.com>

Pentru RPA – UiPath

7. UiPath – UiPath Academy: <https://academy.uipath.com>
8. UiPath Community Forum – <https://forum.uipath.com>
9. Attended Automation for RPA Developers – UiPath Academy
10. Microsoft PowerPoint Automation in StudioX – UiPath Academy
11. Introduction to Automation Cloud Platform – UiPath Academy

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

Conținutul disciplinei este elaborat în concordanță cu direcțiile actuale de cercetare și dezvoltare din domeniul sistemelor distribuite, cloud computing, microservicii și prelucrarea datelor la scară largă; recomandările comunității academice și științifice din domeniul Informaticii și Ingineriei Software; cerințele pieței muncii privind competențele avansate în proiectarea, implementarea și administrarea sistemelor distribuite.

Disciplina contribuie la formarea unor competențe profesionale aliniate cerințelor actuale ale companiilor din domeniul dezvoltării software, administrării infrastructurilor IT, DevOps și cloud computing, facilitând integrarea absolvenților în roluri precum: arhitect software, dezvoltator aplicații distribuite, inginer DevOps, administrator sisteme distribuite sau specialist în baze de date distribuite.

9. Evaluare

Tip de activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
9.1. Curs	Însușirea cunoștințelor teoretice; cunoașterea terminologiei utilizate în domeniu; capacitatea de comunicare utilizând limbaj de specialitate	Examinare scrisă și/sau orală	40%
	Participarea activă la curs, intervenții pertinente, capacitatea de analiză critică	Evaluare continuă	10%
9.2. Laborator	Demonstrarea capacității de analiză, sinteză și aplicare a cunoștințelor teoretice în rezolvarea problemelor și fundamentarea soluțiilor	Evaluare continuă (metode orale și probe practice)	40%

Tip de activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
	Implicarea în rezolvarea aplicațiilor practice; realizarea și susținerea proiectului final de laborator	Evaluare sumativă (pe baza tematicii parcurse în semestru	10%

9.3. Standard minim de performanță

Pentru obținerea notei minime de promovare (5), studentul masterand trebuie să demonstreze următoarele competențe minime:

- Însușirea noțiunilor fundamentale privind programarea orientată pe obiect în Java (clase, obiecte, metode, constructori);
- Înțelegerea mecanismelor de tratare a excepțiilor și aplicarea lor într-o aplicație simplă;
- Capacitatea de a dezvolta o aplicație Java de bază (consolă sau interfață grafică simplă);
- Utilizarea elementară a conexiunii la baze de date prin JDBC;
- Înțelegerea principiilor de bază ale automatizării proceselor (RPA);
- Realizarea unei aplicații simple de automatizare utilizând UiPath Studio sau implementarea unui mini-proiect Java funcțional;
- Utilizarea corectă a terminologiei de specialitate;
- Obținerea a minimum 50% din punctajul total al evaluării.

Neîndeplinirea cerințelor minime la componenta practică (laborator/proiect) conduce la nepromovarea disciplinei, indiferent de rezultatul obținut la evaluarea teoretică.

Data completării

20.09.2025

Titular de curs

Conf.univ.dr. Florentina-Anica PINTEA

Data avizării în departament

26.09.2025

Director de departament

Conf.univ.dr. Victoria IORDAN