

**FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ
DEPARTAMENTUL INGINERIA SOFTWARE ȘI AUTOMATICĂ**

APROBAT

la ședința Departamentului ISA

nr. _____ din _____

Șef departament

FIODOROV Ion

conf. univ., dr.

APROBAT

la ședința Consiliului Facultății CIM

nr. _____ din _____

Președintele Consiliului CIM

CIORBĂ Dumitru,

conf. univ., dr.

Program de studii: *0714.6 Automatică și Informatică*

Cod, Denumirea modului: *S.O.010 Programarea independentă de platformă*

Beneficiari: *Studenții anului IV, învățământ cu frecvență*

Ciclul de învățământ: *Studii superioare de licență, ciclul I*

Numărul de credite ECTS: *4*

Titularul disciplinei

lect. univ. MORARU Dumitru

Nume, prenume, semnătura titularului

I. PRELIMINARII

Disciplina vizează principiile de realizare a programelor independente de platformă.

Scopul disciplinei este de a învăța studenții cum să dezvolte aplicații software care să funcționeze pe mai multe platforme hardware și sisteme de operare, fără a fi nevoie de modificări ale codului sursă. Aceasta înseamnă că programul va putea fi rulat pe diverse medii, cum ar fi Windows, macOS, Linux, sau chiar pe dispozitive mobile, precum Android și iOS.

Obiectivele disciplinei includ:

1. Cunoașterea principiilor de abstractizare. Studenții vor învăța cum să izoleze codul dependent de platformă și să utilizeze interfețe și biblioteci care să permită o portabilitate ridicată.

2. Folosirea tehnologiilor cross-platform. Familiarizarea cu limbaje de programare și framework-uri care suportă programarea independentă de platformă, cum ar fi Java (JVM), Python, C++ (cu librării cum ar fi Qt), sau alte soluții moderne, precum Electron sau Flutter.

3. Gestionarea compatibilității între platforme. Testarea de aplicații pe diverse sisteme de operare, având grijă de aspecte precum diferitele arhitecturi hardware sau modurile de interacțiune cu utilizatorul.

4. Dezvoltarea abilităților de gândire adaptive. Abordarea unor provocări specifice legate de compatibilitatea codului și gestionarea bibliotecilor și resurselor externe, astfel încât programul să fie cât mai flexibil și robust.

Prin această disciplină, studenții își vor dezvolta capacitatea de a construi soluții software scalabile și eficiente, capabile să ruleze pe multiple dispozitive, reducând astfel timpul și efortul necesar adaptării aplicațiilor pentru medii variate.

II. PRECONDIȚII DE ACCES LA UNITATEA DE CURS/MODUL

Pentru a atinge obiectivele cursului studenții trebuie să posede cunoștințe obținute de la disciplinele predecesoare: Matematica, Fizica, Programarea Calculatoarelor, SDA, Programarea Orientată pe Obiect. Legături interdisciplinare cu disciplinele studiate concomitent: Analiza matematică, Programarea calculatoarelor, Grafica pe calculator, Sisteme cu microprocesoare, Arhitecturi de calculatoare, Sisteme de operare.

III. COMPETENȚELE CARE URMEAZĂ A FI DEZVOLTATE

Competențele obținute de această unitate de curs vor fi utilizate în cadrul studierii disciplinelor de specialitate, proiectării de licență și proiectării sistemelor de reglare automată.

Unitatea de curs prevede formarea următoarelor competențe profesionale și transversale:

CP1. Utilizarea de cunoștințe de matematică, fizică, tehnica măsurării, grafică inginerescă, mecanică, electrică și electronică, în ingineria sistemelor.

Utilizarea în comunicarea profesională a conceptelor, teoriilor și metodelor științelor fundamentale folosite în ingineria sistemelor.

Explicarea temelor de rezolvat și argumentarea soluțiilor din ingineria sistemelor, prin utilizarea tehnicilor, conceptelor și principiilor din matematică, fizică, grafică inginerescă, inginerie electrică, electronică.

Rezolvarea problemelor uzuale din domeniul ingineriei sistemelor prin identificarea de tehnici, principii, metode adecvate și prin aplicarea matematicii, cu accent pe metodele de calcul numeric.

Aprecierea potențialului, avantajelor și dezavantajelor unor metode și procedee din domeniul ingineriei sistemelor, a nivelului de documentare științifică al proiectelor și al consistenței aplicațiilor folosind tehnici matematice și alte metode științifice.

Elaborarea de proiecte în domeniul ingineriei sistemelor, selectând și aplicând metode matematice și alte metode științifice specific domeniului.

CP3. Utilizarea fundamentelor automatizării, a metodelor de modelare, simulare, identificare și analiză a proceselor, a tehnicilor de proiectare asistată de calculator.

Identificarea conceptelor fundamentale ale teoriei sistemelor, ingineriei reglării automate, a principiilor de bază din modelare și simulare, precum și a metodelor de analiză a proceselor, în scopul explicării problemelor de bază din domeniu.

Explicarea și interpretarea problemelor de automatizare a unor tipuri de procese prin aplicarea fundamentelor automatizării, a metodelor de modelare, identificare, simulare și analiza proceselor, precum și a tehnicilor de proiectare asistată de calculator.

Rezolvarea unor tipuri de probleme de conducere prin: folosirea de metode și principii de modelare, elaborarea de scenarii de simulare, aplicarea de metode de identificare și de analiză a unor procese (inclusiv procese tehnologice) și sisteme.

Evaluarea performanțelor sistemelor automate, a punctelor tari și punctelor slabe (analiza SWOT) ale proiectelor, a consistenței metodelor și fundamentărilor teoretice.

Configurarea și implementarea sistemelor de conducere a proceselor industriale, roboților și liniilor de fabricație flexibile, precum și alegerea echipamentelor, acordarea și punerea în funcțiune a structurilor aferente.

CP4. Proiectarea, implementarea, testarea, utilizarea și mentenanța sistemelor cu echipamente de uz general și dedicat, inclusiv rețele de calculatoare, pentru aplicații de automată și informatică aplicată.

Definirea cu ajutorul principiilor de funcționare și proiectare, a cerințelor standardelor aplicabile și a metodelor de implementare, testare, mentenanță și exploatare a echipamentelor folosite în aplicațiile de automată și informatică aplicată.

Explicarea și interpretarea metodelor de proiectare, implementare, testare, utilizare și mentenanță a echipamentelor de uz general și dedicat, folosite pentru aplicații de conducere automată și de informatică aplicată.

Rezolvarea de probleme practice de monitorizare și conducere automată și de probleme de informatică aplicată prin utilizarea și adaptarea de echipamente (analogice și numerice) și prin folosirea de tehnologii informatice.

Evaluarea prin monitorizare, diagnoză, analiză de date experimentale, în concordanță cu standarde specifice de performanță a activităților de proiectare, implementare, testare-validare, exploatare și mentenanță a echipamentelor și rețelelor de calculatoare folosite pentru conducere automată și aplicații de informatică.

CT1. Aplicarea, în contextul respectării legislației, a drepturilor de proprietate intelectuală (inclusiv transfer tehnologic), a metodologiei de certificare a produselor, a principiilor, normelor și valorilor codului de etică profesională în cadrul propriei strategii de muncă riguroasă, eficientă și responsabilă.

CT2. Identificarea rolurilor și responsabilităților într-o echipă plurispecializată, luarea deciziilor și atribuirea de sarcini, cu aplicarea de tehnici de relaționare și muncă eficientă în cadrul echipei.

CT3. Identificarea oportunităților de formare continuă și valorificarea eficientă a resurselor și tehnicilor de învățare pentru propria dezvoltare.

IV. ADMINISTRAREA UNITĂȚII DE CURS

Cod	Anul	Semestrul	Numărul de ore						Credite
			Curs	Seminar	Lucrări de laborator	Lucrări practice	Proiectare	Lucrul individual	
S.O.010	Învățământ cu frecvență								
	IV	7	30			30		60	4

V. REZULTATELE ÎNVĂȚĂRII, CONȚINUTURI ȘI METODE DIDACTICE APLICATE

Rezultatele învățării. Studentul trebuie:	Conținuturi		Metode de predare	Realizarea în timp (ore)	
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență	
			curs	lab	
1	2	3	4	5	6
<ul style="list-style-type: none"> • să cunoască noțiuni de bază despre diferențele dintre platformele de operare și cum funcționează mașinile virtuale (ex. JVM). Familiaritate cu concepte de abstractizare a hardware-ului și middleware-ului. • să înțeleagă principiul "write once, run anywhere", avantajele programării multi-platformă și limitările acestui model. • să fie capabil să configureze medii de dezvoltare multi-platformă, să scrie cod care poate fi executat pe mai multe platforme fără modificări. 	T1. Introducere în programarea independentă de platformă.	LP1. Inițiere în limbajul de programare, configurarea mediului de lucru.	<ul style="list-style-type: none"> • expunerea interactivă • prezentări • exemplificare • învățare prin colaborare 	2	2
<ul style="list-style-type: none"> • să cunoască setul de cuvinte cheie, regulile de numire a identificatorilor, tipurile de operatori și separatori folosiți în Java. Diferența dintre expresii și instrucțiuni. • să înțeleagă structura și regulile de formare a unui program Java, inclusiv utilizarea corectă a comentariilor și delimitatorilor. • să fie capabil să scrie programe care respectă regulile sintactice și lexicale ale limbajului, să utilizeze corect operatorii și expresiile. 	T2. Structura lexicală a limbajului Java.		<ul style="list-style-type: none"> • expunerea interactivă • prezentări • exemplificare • învățare prin colaborare 	2	
<ul style="list-style-type: none"> • să cunoască tipuri primitive (int, float, char, boolean etc.) și referențiale (clase, obiecte, tablouri) în Java. Structurile de control (if, else, switch, for, while) și conceptul de vector (tablouri unidimensionale și multidimensionale). • să înțeleagă diferențele între tipuri primitive și referențiale, manipularea corectă a vectorilor și controlul execuției în cadrul programelor. • să fie capabil să declare și utilizeze variabile și vectori, să implementeze structuri de control pentru a gestiona logica programelor și să manipuleze tablouri eficient. 	T3. Tipuri de date. Controlul execuției. Vectori.	LP2. Tipuri de date și controlul execuției.	<ul style="list-style-type: none"> • expunerea interactivă • prezentări • exemplificare • învățare prin colaborare 	2	2
<ul style="list-style-type: none"> • să cunoască principiile programării orientate pe obiecte (OOP): clase, obiecte, metode, constructori, moștenire, încapsulare și polimorfism. 	T4. Obiecte și clase în Java. Organizarea claselor.		<ul style="list-style-type: none"> • expunerea interactivă • prezentări 	2	

Rezultatele învățării. Studentul trebuie:	Conținuturi		Metode de predare	Realizarea în timp (ore)	
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență	
				curs	lab
1	2	3	4	5	6
<ul style="list-style-type: none"> să înțeleagă cum se organizează codul în clase, relațiile dintre clase (moștenire, compoziție) și utilizarea polimorfismului pentru abstractizarea comportamentului. să fie capabil să creeze și să organizeze clase și obiecte, să implementeze conceptele de moștenire și polimorfism și să gestioneze instanțierea și utilizarea corectă a obiectelor. 			<ul style="list-style-type: none"> exemplificare învățare prin colaborare 		
<ul style="list-style-type: none"> să cunoască bibliotecile de bază pentru dezvoltarea interfețelor grafice în Java (AWT, Swing), componentele UI (ferestre, butoane, panouri) și evenimentele utilizatorului. să înțeleagă cum să construiască și să organizeze elementele unei interfețe grafice și să gestioneze evenimentele (ex. clicuri de buton). să fie capabil să creeze ferestre și elemente interactive de UI, să gestioneze interacțiunea utilizatorului cu aplicația și să implementeze logica evenimentelor. 	T5. Interfața grafică cu utilizatorul.	LP3. Obiecte și clase în Java.	<ul style="list-style-type: none"> expunerea interactivă prezentări exemplificare învățare prin colaborare 	6	2
<ul style="list-style-type: none"> să cunoască modelul de execuție concurentă în Java (thread-uri), clasele și metodele Java pentru gestionarea firelor de execuție, sincronizarea și managementul competiției între fire. să înțeleagă cum se creează și gestionează firele de execuție, problemele de competiție între fire și modul de sincronizare corectă a acestora. să fie capabil să creeze și să gestioneze fire de execuție multiple, să folosească mecanisme de sincronizare și să evite problemele de acces concurent la resurse. 	T6. Fire de execuție	LP4. Interfața grafică cu utilizatorul.	<ul style="list-style-type: none"> expunerea interactivă prezentări exemplificare învățare prin colaborare 	4	8
<ul style="list-style-type: none"> să cunoască noțiuni de bază despre rețele și protocoale (TCP/IP), utilizarea socket-urilor în Java pentru comunicarea prin rețea. să înțeleagă modelul client-server și cum se gestionează comunicarea între mașini prin rețea, utilizând socket-uri. 	T7. Programare în rețea		<ul style="list-style-type: none"> expunerea interactivă prezentări exemplificare 	4	

Rezultatele învățării. Studentul trebuie:	Conținuturi		Metode de predare	Realizarea în timp (ore)	
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență	
			curs	lab	
1	2	3	4	5	6
<ul style="list-style-type: none"> să fie capabil să scrie aplicații care se conectează la rețea, să implementeze atât partea de client, cât și de server și să gestioneze comunicarea prin socket-uri. 			<ul style="list-style-type: none"> învățare prin colaborare 		
<ul style="list-style-type: none"> să cunoască conceptul de baze de date relaționale, interogări SQL și conectivitatea la baze de date în Java folosind JDBC. să înțeleagă cum se conectează și interacționează aplicațiile Java cu o bază de date, folosind interogări SQL pentru a insera, actualiza și extrage date. să fie capabil să configureze conexiuni la baze de date, să execute interogări SQL și să manipuleze datele extrase din baza de date. 	T8. Lucrul cu baza de date.	LP5. Aplicație simplă client-server folosind socket-uri.	<ul style="list-style-type: none"> expunerea interactivă prezentări exemplificare învățare prin colaborare 	4	8
<ul style="list-style-type: none"> să cunoască API-ul de reflecție din Java, care permite descoperirea și manipularea caracteristicilor claselor la runtime. să înțeleagă cum să încarce clase dinamic, să acceseze metode și câmpuri la runtime și să creeze instanțe ale claselor într-un mod flexibil. să fie capabil să folosească reflecția pentru a manipula clase și obiecte la runtime, să încarce clase dinamice și să acceseze metode și câmpuri dinamic. 	T9. Lucrul dinamic cu clase.	LP6. Achiziția datelor de la dispozitiv incorporat.	<ul style="list-style-type: none"> expunerea interactivă prezentări exemplificare învățare prin colaborare 	4	8

VI. SUGESTII PENTRU ACTIVITATEA INDIVIDUALĂ A STUDENȚILOR

Pe parcursul semestrului, studenții realizează activități individuale, care includ:

- studiul literaturii obligatorii conform listei surselor bibliografice prezentate în curriculum;
- realizarea temelor pentru acasă, propuse în cadrul lucrărilor de laborator;

VII. UTILIZAREA INTELIGENȚEI ARTIFICIALE GENERATIVE

Permișunea de utilizare:

Utilizarea IA generative în cadrul temelor și proiectelor este permisă, cu condiția ca studenții să respecte următoarele reguli:

IA generativă poate fi utilizată pentru generarea de idei, structuri de text sau cod, dar toate materialele generate trebuie să fie revizuite și ajustate de către student pentru a se asigura că acestea corespund cerințelor academice.

Orice utilizare a IA generative trebuie să fie declarată în secțiunea de apendice a fiecărei lucrări, folosind fraza: "În timpul pregătirii acestei lucrări, autorul a utilizat [NUME INSTRUMENT / SERVICIU] în scopul [MOTIV]. După utilizarea acestui instrument/serviciu, autorul a revizuit și editat conținutul după cum a fost necesar și își asumă întreaga responsabilitate pentru conținutul lucrării."

Restricții de utilizare:

Studenții nu trebuie să considere IA generativă ca o sursă de încredere pentru informații, deoarece nu oferă referințe clare sau surse documentate.

Nu este permisă citarea directă a conținutului generat de IA în lucrările academice ca și cum ar fi sursă primară.

Activitățile în care este interzis utilizarea IA generativă sunt specificare de profesor și sunt de regulă evaluări intermediare și finale sau care nu presupun activități de dezvoltare a competențelor profesionale.

VIII. EVALUAREA DISCIPLINEI

Periodică		Curentă	Studiu individual	Proiect/teză	Examen
EP 1	EP 2				
Învățământ cu frecvență					
15%	15%	15%	15%		40%
Standard minim de performanță Prezența și activitatea la prelegeri și lucrări de laborator; Obținerea notei minime de „5” la fiecare dintre atestări și lucrări de laborator; Demonstrarea în lucrarea de examinare finală a cunoașterii proceselor și tehnologiilor de bază aplicate la programarea independentă de platformă.					

VIII. CRITERII DE EVALUARE

Activitate	Componente evaluare	Metodă de evaluare, Criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
Învățământ cu frecvență				
Evaluare periodică I	Conținut teoretic, teme 1-5	Studentul primește o sarcină de lucru în care va realiza un program simplificat independent de platformă.	100%	15%
Evaluare periodică II	Conținut teoretic, teme 6-9	Studentul primește o sarcină de lucru în care va realiza un program independent de platformă cu interfață grafică.	100%	15%

Activitate	Componente evaluare	Metodă de evaluare, Criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
Evaluare curentă	Activitatea practică	Discuții în cadrul orelor practice	50%	15%
		Dosar completat cu Rapoarte pentru fiecare lucrare practică	50%	
Studiul individual	Cercetare la temă	Referat/Prezentare/discurs public	100%	15%
Evaluarea finală	Conținut teoretic și practic	La calculator, în baza biletului individual.	100%	40%

IX. LISTA DE SUBIECTE PENTRU EVALUĂRI PERIODICE ȘI CEA FINALĂ

CHESTIONAR PENTRU EP I

- T1. Introducere în programarea independentă de platformă.
- T2. Structura lexicală a limbajului Java.
- T3. Tipuri de date. Controlul execuției. Vectori.
- T4. Obiecte și clase în Java. Organizarea claselor.
- T5. Interfața grafică cu utilizatorul.

CHESTIONAR PENTRU EP II

- T6. Fire de execuție
- T7. Programare în rețea
- T8. Lucrul cu baza de date.
- T9. Lucrul dinamic cu clase.

CHESTIONAR PENTRU EXAMEN

- T1. Introducere în programarea independentă de platformă.
- T2. Structura lexicală a limbajului Java.
- T3. Tipuri de date. Controlul execuției. Vectori.
- T4. Obiecte și clase în Java. Organizarea claselor.
- T5. Interfața grafică cu utilizatorul.
- T6. Fire de execuție
- T7. Programare în rețea
- T8. Lucrul cu baza de date.
- T9. Lucrul dinamic cu clase.

X. REFERINȚE BIBLIOGRAFICE

Obligatorii

1. FRĂSINARU, C. Curs practic de Java. București, Matrix-Rom, 2005, 223 p. Disponibil: http://thor.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf.
2. HORSTMANN, C. S., CORNELL, G. *Core Java Volume I, Fundamentals*, 11th Edition, Prentice Hall, 2018, 843 p. ISBN 978-0-13-516630-7.
3. SCHMITT, C, SIMPSON, K. *HTML5 Cookbook - Solutions and Examples for HTML5 Developers*. O'Reilly 2012, 260p. ISBN 978-1-449-39679-4.

Suplimentare

4. DEITEL, P., DEITEL, H. *Java How to Program*, 10th Edition, Pearson, 2014, 1248 p. ISBN 978-0-133-80780-6.
5. KURNIAWAN, B. *Java 7: A Beginner's Tutorial*, Third Edition, Brainy Software, 2012, 716p. ISBN 978-0-980-83961-6.