

FISA DISCIPLINEI

PROGRAMAREA CALCULATOARELOR (LIMBAJUL JAVA) COBF204

Număr credite5

1. Obiectivele disciplinei

Constau în însușirea conceptelor de bază ale programării orientate pe obiecte și aplicarea acestora în limbajul de programare Java. Dezvoltarea cunoștințelor pentru crearea de aplicații în limbajul Java, ceea ce implică studierea sintaxei și semanticii limbajului Java, a interfeței de programe de aplicații a limbajului (Java API), a interfeței grafice utilizator și a programării orientate pe evenimente. Însușirea cunoștințelor necesare în utilizarea fluxurilor de date și fișierelor în limbajul Java și a principiilor programării concurente.

2. Rezultatele învățării (*se exprima în obiective măsurabile ce fac subiectul evaluării*)

a. Cunoștințele generale

Introducere în programarea orientată pe obiecte. Identificatori, cuvinte cheie și cuvinte rezervate, literali, separatori și operatori, spații și comentarii. Declararea și inițializarea variabilelor. Tipuri de date primitive și declarații de tip. Expresii cu date primitive. Principiile programării structurate. Structurile de control fundamentale. Structura secvențială. Structura alternativă. Structura de comutare. Structuri repetitive. Instrucțiuni etichetate. Tablouri (tablouri unidimensionale; tablouri multidimensionale). Metode recursive. Tipul referință. Clase, câmpuri, metode. Caracteristici ale obiectelor și claselor (încapsularea; agregarea; moștenirea; polimorfismul). Interfețe. Interfețe utilizator grafice. Componente ale interfeței utilizator grafice. Fluxuri de intrare/ieșire. Fișiere. Operații cu fișiere. Conceptul de proces. Procese paralele și concurente.

b. Cunoștințele de specialitate

Caracteristicile limbajului Java. Principalele tipuri de produse software care se pot scrie în limbajul Java. Șablonul unei aplicații în limbajul Java. Editarea, compilarea și execuția aplicațiilor Java. Unități lexicale ale limbajului Java. Tratarea excepțiilor în limbajul Java. Ierarhia de clase Java. Pachetele de clase din Java API. Utilizarea claselor din pachetul java.lang. Utilizarea parametrilor din linia de comandă. Declararea claselor. Declararea câmpurilor. Constructori. Declararea metodelor. Signatura metodei. Transferul parametrilor către metode. Tratarea excepțiilor. Colectorul de reziduuri. Clase abstracte. Declararea interfețelor. Clase imbricate și clase interioare. Clasele de lucru în interfața grafică utilizator. Programarea orientată pe evenimente. Clasele de bază ale ierarhiilor de fluxuri de intrare/ieșire. Operații cu fișiere. Operații cu fluxuri de date. Fire de execuție.

c. Competențele generale

Studentii trebuie să-și însușească limbajul specific domeniului programării calculatoarelor, să posede competențe esențiale în rezolvarea de probleme, aplicând gestiunea corespunzătoare a timpului, capacități de mobilizare, capacități de comunicare și de lucru în echipă.

d. Competențele de specialitate

Studentii trebuie să fie capabili să identifice și să aplice principiile de programare în rezolvarea de probleme; să înțeleagă și să adapteze regulile programării pe obiecte și pe evenimente corespunzător aplicațiilor în limbajul Java.

e. Abilitățile cognitive specifice

Necesare pentru acest curs sunt eficacitatea personală a studenților prin adaptarea la noi situații, capacitatea de gestionare de situații, pragmatismul și rigurozitatea de care dau dovadă studenții în rezolvarea de aplicații specifice, capacitatea de interacțiune și comunicare.

3. Concordanța cu obiectivele planului de învățământ/specializării

a. Contribuția rezultatelor învățării disciplinei la formarea competențelor specializării

Studentii vor fi deține cunoștințe și abilități specifice limbajelor, mediilor și tehnicilor de programare orientată obiect.

Cunoștințele pe care le vor deține le vor permite să analizeze și să identifice paradigmele de programare orientată obiect, să modeleze și să implementeze principiile de programare structurată.

Studentii vor fi capabili să întocmească și să gestioneze proiecte în domeniul informaticii aplicate folosind programarea în limbajul Java.

Cunoștințele acumulate la acest curs sunt utile la cursurile de Algoritmi și tehnici de programare avansate și Baze de date din anul 2 semestrul 2.

b. Cerințele disciplinare prealabile

Studentii trebuie să dețină cunoștințe generale referitoare la caracteristicile programării procedurale, modul de lucru al limbajelor compilate, tipuri și structuri de date fundamentale sau definite de utilizator, logică matematică, algebră liniară și analiză matematică. Disciplinele de precedentă considerate utile pentru acest curs sunt Programarea calculatoarelor și Limbaje de programare orientate pe obiecte.

4. Structura activității didactice

CURS	28 ore
Seminar	0 ore
Lucrări practice	14 ore
Proiect	14 ore

5. Prezentarea conținutului disciplinei

a. Curs

<i>Conținutul activității</i>	<i>Nr. de ore</i>
Ce este Java; Caracteristicile limbajului Java; Introducere în programarea orientată pe obiecte; Prezentarea principalelor tipuri de produse software care se pot scrie în limbajul Java; Șablonul unei aplicații simple în limbajul Java; Editarea, compilarea și execuția aplicațiilor Java.	2
Comentarii în fișierul sursă; Lucrul cu șiruri de caractere; Unități lexicale ale limbajului Java; Identificatori, cuvinte cheie și cuvinte rezervate, literal, separatori și operatori, spații și comentarii. Declararea și inițializarea variabilelor; Tipuri de date primitive și declarații de tip; Tipul boolean; Tipuri numerice (tipuri de date întregi, tipuri de date în virgulă mobilă, tipul char); Expresii cu date primitive; Expresia condițională; Principiile programării structurate; Structurile de control fundamentale; Blocul ca realizare a structurii secvențiale; Structura alternativă (if .. else); Structura de comutare (switch); Structuri repetitive (ciclul cu test inițial –while; ciclul cu test final -do .. while; ciclul for); Instrucțiuni etichetate; Tratarea excepțiilor în limbajul Java (try .. catch)	2
Tipul referință; Clase, câmpuri, metode; Constructori; crearea obiectelor prin operatorul new; Moștenirea și polimorfismul; Ierarhia de clase Java; Pachetele de clase din Java API; Declarația import;	2
Utilizarea claselor din pachetul java.lang (clasa Object, clasele de excepții, clasa String, clasa Class); Tablouri (tablouri unidimensionale; tablouri multidimensionale); Utilizarea parametrilor din linia de comandă.	2
Declararea claselor; Declararea câmpurilor (câmpuri statice și câmpuri ale instanței); Declararea metodelor (metode statice și metode ale instanței); Signatura metodei; Transferul parametrilor către metode la invocarea metodelor; Metode care întorc o referință la un obiect construit în corpul lor; Metode care au ca argumente și/sau ca valori întoarse referințe la tablouri; Metode recursive; Metode care generează excepții; Instrucțiunea throw și clauza throws; Distrugerea obiectelor de către colectorul de reziduuri; Metoda finalize.	2
Caracteristici ale obiectelor și claselor (încapsularea; câmpuri și metode publice și private; modificatori de acces pentru câmpuri și metode; declararea constructorilor; agregarea; moștenirea; membrii protejați ai clasei; referințele this și super; declararea clasei derivate; clauza extends; declararea constructorului clasei derivate; redefinirea metodelor; ascunderea (acoperirea) câmpurilor; ascunderea metodelor statice; metode finale; declararea propriilor clase de excepții; clase finale; polimorfismul); Clase abstracte;	2
Interfețe; Declararea interfetelor; Clase imbricate și clase interioare;	2
Interfețe utilizator grafice; Clasa Component; Programarea orientată pe evenimente; Clase de	2

evenimente; Clase pentru interfețe utilizator grafice (clasa BorderLayout, clasa FlowLayout, clasa GridLayout, clasa BoxLayout, clasele JPanel și Jpanel, clasa Box).	
Componente ale interfeței utilizator grafice; Componente de control (butoane, casete de validare, butoane radio, liste, liste ascunse, câmpuri de text, rigle cu cursor, meniuri); Componente de afișare (componente de afișare needitabile, componente de afișare editabile).	2
Utilizarea ferestrelor de dialog; Alegerea culorii; Elemente de grafică; Contextul grafic;	2
Fluxuri de intrare/ieșire; Clasele de bază ale ierarhiilor de fluxuri de intrare/ieșire; Considerații generale privind utilizarea fluxurilor;	2
Fișiere; Citirea fișierelor (citirea fișierelor de octeți, citirea fișierelor de caractere); Scrierea în fișiere (scrierea în fișiere de octeți, scrierea în fișiere de caractere); Fișiere cu acces direct; Fluxuri de prelucrare; Fluxuri de date; Fluxuri de obiecte	2
Conceptul de proces; Procese paralele și concurente; Fire de execuție; Sincronizarea firelor de execuție.	2

b. Seminar

Conținutul activității

Nr. de ore

c. Lucrări practice

Conținutul activității

Nr. de ore

Crearea de aplicații simple în limbajul Java; Editarea, compilarea și execuția aplicațiilor Java.	1
Exemple de lucru cu tipurile de date primitive, cu șiruri de caractere. Comentarii în fișierul sursă; Declararea și inițializarea variabilelor;	1
Aplicații de utilizare a structurilor de control fundamentale. Utilizarea expresiilor. Tratarea excepțiilor în limbajul Java.	1
Aplicații cu clase, câmpuri, metode, constructori, crearea de obiecte.	1
Utilizarea claselor din pachetul java.lang. Exemple de utilizare a tablourilor. Utilizarea parametrilor din linia de comandă.	1
Aplicații pentru utilizarea de câmpuri statice și ale instanței, de metode statice și ale instanței, metode cu aceeași semnătură. Folosirea recursiei.	1
Exemple de creare a claselor derivate. Clase abstracte.	1
Aplicații cu interfețe.	1
Interfețe utilizator grafice. Clasa Component. Exemple de utilizare a programării orientată pe evenimente;	1
Utilizarea componentelor în interfața utilizator grafice.	1
Utilizarea ferestrelor de dialog; Alegerea culorii.	1
Aplicații pentru utilizarea fluxurilor.	1
Utilizarea fișierelor.	1
Exemple de folosire a firelor de execuție.	1

d. Proiect

Conținutul activității

Nr. de ore

Prezentarea temei de proiect. Enumerarea cerințelor pentru realizarea proiectului. Alegerea echipelor de lucru la proiect.	2
Analiza temei de proiect și proiectarea modulelor necesare la rezolvare.	2
Implementarea modulelor cerute în realizarea temei de proiect.	8
Prezentarea proiectului realizat.	2

6. Învățare

a. Forme de învățare/predare

Ca forme de predare se folosesc expunerea materialului de curs, care cuprinde aspecte teoretice și exemplificarea acestora în aplicații practice la orele de laborator. Pentru o mai bună înțelegere principiilor programării se realizează exemple comparative. La aplicațiile practice se prezintă o serie de tipuri de probleme rezolvate, iar studenții sunt îndrumați să folosească similaritatea pentru rezolvarea unor probleme asemănătoare.

b. Resurse educaționale

Resursele on-line ale suportului de curs și explicații ale aplicațiilor de laborator prezentate ca model și

ale celor propuse spre rezolvare sunt prezente pe site-ul catedrei, la care au acces studenții.

c. Bibliografie disponibilă

Pagina Web a firmei Sun Microsystems pentru utilizatorii limbajului Java este java.sun.com.

Bumbaru S., Programarea orientată pe obiecte în limbajul Java, Ed. Universitatea "Dunărea de Jos" din Galați, 2002

Athanasiu Irina, Costinescu B., Drăgoi O.A., Popovici F.I. Limbajul Java. O perspectivă pragmatică. Editura Agora, Tg.Mureș

Lemay L., Cadenhead R. Java2 Fără profesor în 21 zile. Editura Teora, București

Chan M.C., Griffith S.W., Iasi A.F. Java - 1001 secrete pentru programatori. Editura Teora, București

d. Alte resurse

Laboratoare cu dotări în calculatoare pe care sunt instalate platforma Java și medii de programare specifice limbajului Java.

7. Studiu individual: 70 ore

8. Evaluare

a. Forme de evaluare

A Pe perioada studiului studenții sunt evaluați diagnostic formativ pentru monitorizarea progresului, a rezultatelor și a nivelului de cunoaștere la care au ajuns. Se asigură astfel urmărirea cunoștințelor dobândite de studenți și eventualele modalități de remediere necesare.

B La sfârșitul semestrului se realizează o evaluare sumativă prin examinare scrisă pentru verificarea cunoștințelor teoretice ale programării calculatoarelor și elaborarea practică și susținerea orală a unor aplicații de programare în limbajul Java.

b. Principii de notare

Notarea la examen constă din 50% la proba scrisă și 50% pentru elaborarea și susținerea aplicațiilor practice.

c. Informarea studenților cu privire la evaluarea asociată disciplinei

Informarea studenților cu privire la modalitatea de desfășurare a predării disciplinei și modalitatea de examinare și notare se realizează la primul curs. De asemenea, studenții sunt atenționați să depună un efort susținut asupra aspectelor aplicative ale programării în limbajul Java.

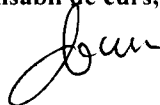
9. Responsabil de disciplina

Nume : Ș.I. Cocu Adina

Date de contact : adina.cocu@ugal.ro, Universitatea „Dunărea de Jos”, Galați, str. Domnească, Corp G, cam. G309.

Programul de contact se va desfășura conform programului afișat la avizierul catedrei.

Responsabil de curs,



Sef Departament / Catedra,

