**Ministerul Educației, Culturii și Cercetării al Republicii Moldova**

**IP Centrul de Excelență în Informatică și Tehnologii Informaționale**

**Catedra de Informatică II**



**Lucrare de studiu individual nr.1**

**Disciplina:** Programarea Orientată pe Obiecte

**Tema:** Proiect individual. Elaborarea claselor în limbajul Java utilizând constructori și destructori

|  |  |
| --- | --- |
| **Elaborat**:  Elevul grupei B-1821  ***Specialitatea:*** Administrarea Bazelor De Date  ***Nume*** *Platon Bogdan* | **Verificat**:  ***Luncasu*** ,  profesor de Programare Orientată pe  Obiecte |

Data prezentării:\_\_\_\_\_\_\_\_\_\_

Nota\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Cod Java

# Clasa Sportiv

import static java.lang.System.\*;  
import java.time.\*;  
import java.time.format.DateTimeFormatter;  
  
class Sportivi  
{  
 String DataNasterii;  
 int cod;  
 double GreutateInKilograme, InaltimeaInCentimetri;  
 String nume, prenume, gen;  
 boolean StudiiSuperioare;  
  
 Sportivi()  
 {  
  
 }Sportivi(int cod ,String nume, String prenume, String gen , String DataNasterii ,  
 double GreutateaInKilograme, double InaltimeaInCentimentri, boolean StudiiSuperioare)  
 {  
 this.nume = nume;  
 this.prenume = prenume;  
 this.gen = gen;  
 this.StudiiSuperioare = StudiiSuperioare;  
 this.DataNasterii = DataNasterii;  
 this.cod = cod;  
 this.GreutateInKilograme = GreutateaInKilograme;  
 this.InaltimeaInCentimetri = InaltimeaInCentimentri;  
 }  
  
 String getNume()  
 {  
 return nume;  
 }  
  
 String getPrenume()  
 {  
 return prenume;  
 }  
  
 String getGen()  
 {  
 return gen;  
 }  
  
 boolean getStudiiSuperioare()  
 {  
 return StudiiSuperioare;  
 }  
  
 LocalDate DataNasterii()  
 {  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("d/MM/yyyy");  
 return LocalDate.*parse*(DataNasterii, formatter);  
 }  
  
 int getCod()  
 {  
 return cod;  
 }  
  
 double getGreutateInKilograme()  
 {  
 return GreutateInKilograme;  
 }  
  
 double getInaltimeaInCentimetri()  
 {  
 return InaltimeaInCentimetri;  
 }  
  
 String getProbaSportiva()  
 {  
 return "default";  
 }  
  
 int getVarsta()  
 {  
 LocalDate now = LocalDate.*now*();  
 Period p = Period.*between*(DataNasterii(), now);  
 int Varsta = p.getYears();  
  
 return Varsta;  
  
 }  
  
 void afisare()  
 {  
 *out*.println("id: " + getCod());  
 *out*.println("Nume: " + getNume());  
 *out*.println("Prenume: " + getPrenume());  
 *out*.println("Data Nasterii: " + DataNasterii());  
 *out*.println("Varsta: " + getVarsta());  
 *out*.println("Gen: " + getGen());  
 *out*.println("Inaltime: " + getInaltimeaInCentimetri() + "cm");  
 *out*.println("Masa: " + getGreutateInKilograme() + "kg");  
 *out*.println("Studii Superioare: " + getStudiiSuperioare());  
 }  
}

# **Clasa Fotal**

import java.time.LocalDate;  
import static java.lang.System.\*;  
  
  
public class Fotbal extends Sportivi  
{  
 String echipa, divizie, pozitie;  
 int goluri, asistari;  
 double pretJucator;  
  
 Fotbal()  
 {  
  
 }Fotbal(int cod ,String nume, String prenume, String gen , String DataNasterii ,  
 double GreutateaInKilograme, double InaltimeaInCentimentri, boolean StudiiSuperioare,  
 String echipa, String divizie, String pozitie, int goluri, int asistari, double pretJucator)  
 {  
 super(cod, nume, prenume, gen, DataNasterii, GreutateaInKilograme, InaltimeaInCentimentri, StudiiSuperioare);  
 this.echipa = echipa;  
 this.divizie = divizie;  
 this.pozitie = pozitie;  
 this.goluri = goluri;  
 this.asistari = asistari;  
 this.pretJucator = pretJucator;  
 }  
  
  
 @Override  
 String getProbaSportiva()  
 {  
 return "Fotbal";  
 }  
  
 String getEchipa()  
 {  
 return echipa;  
 }  
  
 String getDivizie()  
 {  
 return divizie;  
 }  
  
 String getPozitie()  
 {  
 return pozitie;  
 }  
  
 int getGoluri()  
 {  
 return goluri;  
 }  
  
 int getAsistari()  
 {  
 return asistari;  
 }  
  
 double getPretJucator()  
 {  
 return pretJucator;  
 }  
  
 @Override  
 void afisare()  
 {  
 super.afisare();  
 *out*.println("echipa: " + getEchipa());  
 *out*.println("divizie: " + getDivizie());  
 *out*.println("pozitie: " + getPozitie());  
 *out*.println("goluri: " + getGoluri());  
 *out*.println("asistari: " + getAsistari());  
 *out*.println("pretul jucatorului: " + getPretJucator() + " $");  
 *out*.println("============================================");  
 }  
}

# **Clasa Basketball**

import java.time.LocalDate;  
import static java.lang.System.\*;  
  
public class Basketball extends Sportivi  
{  
 String echipa ,divizie, liga, pozitie;  
 int MarcariTreiPuncte, MarcariDouaPuncte;  
 double pretJucator;  
  
 Basketball()  
 {  
  
 }Basketball(int cod ,String nume, String prenume, String gen , String DataNasterii ,  
 double GreutateaInKilograme, double InaltimeaInCentimentri, boolean StudiiSuperioare, String echipa ,  
 String divizie, String liga, String pozitie, int MarcariTreiPuncte, int MarcariDouaPuncte, double pretJucator)  
 {  
 super(cod, nume, prenume, gen, DataNasterii, GreutateaInKilograme, InaltimeaInCentimentri, StudiiSuperioare);  
 this.echipa = echipa;  
 this.divizie = divizie;  
 this.liga = liga;  
 this.pozitie = pozitie;  
 this.MarcariDouaPuncte = MarcariDouaPuncte;  
 this.MarcariTreiPuncte = MarcariTreiPuncte;  
 this.pretJucator = pretJucator;  
 }  
  
 @Override  
 String getProbaSportiva()  
 {  
 return "Basket";  
 }  
  
 String getDivizie()  
 {  
 return divizie;  
 }  
  
 String getLiga()  
 {  
 return liga;  
 }  
  
 String getPozitie()  
 {  
 return pozitie;  
 }  
  
 int getMarcariDouaPuncte()  
 {  
 return MarcariDouaPuncte;  
 }  
  
 int getMarcariTreiPuncte()  
 {  
 return MarcariTreiPuncte;  
 }  
  
 double getPretJucator()  
 {  
 return pretJucator;  
 }  
  
 @Override  
 void afisare()  
 {  
 super.afisare();  
 *out*.println("Divizie: " + getDivizie());  
 *out*.println("Liga: " + getLiga());  
 *out*.println("Pozitie: " + getPozitie());  
 *out*.println("Marari 2 puncte: " + getMarcariDouaPuncte());  
 *out*.println("Marcari 3 puncte: " + getMarcariTreiPuncte());  
 *out*.println("Pretul Jucatorului: " + getPretJucator());  
 *out*.println("============================================");  
  
 }  
}

# **Clasa Tenis**

import java.time.LocalDate;  
import static java.lang.System.\*;  
  
public class Tenis extends Sportivi  
{  
 int rank, recordPierderi, recordVictorii;  
 String divizie;  
  
 Tenis()  
 {  
  
 }Tenis(int cod ,String nume, String prenume, String gen , String DataNasterii ,  
 double GreutateaInKilograme, double InaltimeaInCentimentri, boolean StudiiSuperioare,  
 int rank, String divizie, int recordVictorii, int recordPierderi)  
 {  
 super(cod, nume, prenume, gen, DataNasterii, GreutateaInKilograme, InaltimeaInCentimentri, StudiiSuperioare);  
 this.rank = rank;  
 this.divizie = divizie;  
 this.recordVictorii = recordVictorii;  
 this.recordPierderi = recordPierderi;  
  
 }  
  
 @Override  
 String getProbaSportiva()  
 {  
 return "tenis";  
 }  
  
 int getRank()  
 {  
 return rank;  
 }  
  
 String getDivizie()  
 {  
 return divizie;  
 }  
  
 int getRecordVictorii()  
 {  
 return recordVictorii;  
 }  
  
 int getRecordPierderi()  
 {  
 return recordPierderi;  
 }  
  
 @Override  
 void afisare()  
 {  
 super.afisare();  
 *out*.println("rank : " + rank);  
 *out*.println("divizie: " + divizie);  
 *out*.println("victorii: " + recordVictorii);  
 *out*.println("infringeri: " + recordPierderi);  
 *out*.println("============================================");  
 }  
}

# **Clasa Inot**

class Inot extends Sportivi  
{  
 int rank;  
 String divizie, categorie, nationala;  
  
 Inot()  
 {  
  
 }Inot(int cod ,String nume, String prenume, String gen , String DataNasterii ,  
 double GreutateaInKilograme, double InaltimeaInCentimentri, boolean StudiiSuperioare,int rank, String divizie, String categorie, String nationala)  
 {  
 super(cod, nume, prenume, gen, DataNasterii, GreutateaInKilograme, InaltimeaInCentimentri, StudiiSuperioare);  
 this.rank = rank;  
 this.divizie = divizie;  
 this.categorie = categorie;  
 this.nationala = nationala;  
 }  
  
 @Override  
 String getProbaSportiva()  
 {  
 return "inot";  
 }  
  
 int getRank()  
 {  
 return rank;  
 }  
  
 String getDivizie()  
 {  
 return divizie;  
 }  
  
 String getCategorie()  
 {  
 return categorie;  
 }  
  
 String getNationala()  
 {  
 return nationala;  
 }  
}

# **Clasa FotbalAfisareCitire**

import java.util.InputMismatchException;  
import java.util.Scanner;  
import java.util.ArrayList;  
import static java.lang.System.*out*;  
  
/\*  
  
clasa folosita pentru citirea si afisarea datelor de la tastatura  
 pentru tipul de date fotbalist  
\*/  
  
  
//citirea tipului de date fotbalisti care vor fi introdusi in array-ul dinamic sportivi unde se vor afla si alte tipuri de date  
  
class FotbalistiAfisareCitire {  
 public static void CitireaFotbalisti(ArrayList<Sportivi> Sportivi, Scanner scanner, int n) throws InputMismatchException {  
 try {  
 for (int i = 0; i < n; i++) {  
 *out*.print("cod: ");  
 int cod = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Nume: ");  
 String nume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Prenume: ");  
 String prenume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("gen: ");  
 String gen = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Inaltimea: ");  
 double InaltimeaInCentimentri = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Masa: ");  
 double GreutateaInKilograme = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Data Nasterii: MM/DD/YY");  
 String dataNasterii = scanner.next();  
  
 *out*.println("\n");  
  
  
 *out*.print("Studii Superioare: ");  
 boolean StudiiSuperioare = scanner.hasNext();  
  
 *out*.print("echipa: ");  
 String echipa = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("divizie: ");  
 String divizie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("pozitie: ");  
 String pozitie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("goluri: ");  
 int goluri = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("asistari: ");  
 int assistari = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("pretul jucatorului: ");  
 double pretJucator = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 Sportivi.add(new Fotbal(cod, nume, prenume, gen, dataNasterii, GreutateaInKilograme, InaltimeaInCentimentri,  
 StudiiSuperioare, echipa, divizie, pozitie, goluri, assistari, pretJucator));  
 }  
 }catch (java.util.InputMismatchException e)  
 {  
 scanner.next();  
 *out*.println("Tipul de date este introdus gresit");  
 }  
 }  
//afisarea datelor citite  
  
 public static void Afisare(ArrayList<Sportivi> Sportivi)  
 {  
 for(int i = 0; i < Sportivi.size(); i++)  
 {  
 Sportivi.get(i).afisare();  
 }  
 }  
}

# Clasa BasketAfisareCitire

import java.util.InputMismatchException;  
import java.util.Scanner;  
import java.util.ArrayList;  
import static java.lang.System.*out*;  
  
/\*  
  
clasa folosita pentru citirea si afisarea datelor de la tastatura  
 pentru tipul de date basket  
\*/  
  
//citirea tipului de date basket care vor fi introdusi in array-ul dinamic sportivi unde se vor afla si alte tipuri de date  
class BasketAfisareCitire  
{  
 public static void CitireaBasketball(ArrayList<Sportivi> Sportivi, Scanner scanner, int n) throws InputMismatchException {  
 try {  
 for (int i = 0; i < n; i++) {  
 *out*.print("cod: ");  
 int cod = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Nume: ");  
 String nume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Prenume: ");  
 String prenume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("gen: ");  
 String gen = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Inaltimea: ");  
 double InaltimeaInCentimentri = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Masa: ");  
 double GreutateaInKilograme = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Data Nasterii: MM/DD/YY");  
 String dataNasterii = scanner.next();  
  
 *out*.println("\n");  
  
  
 *out*.print("Studii Superioare: ");  
 boolean StudiiSuperioare = scanner.hasNext();  
  
 *out*.print("echipa: ");  
 String echipa = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("divizie: ");  
 String divizie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("liga: ");  
 String liga = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("pozitie: ");  
 String pozitie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Marcari2Puncte: ");  
 int Marcari2Puncte = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Marcari3Puncte: ");  
 int Marcari3Puncte = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Pret jucator: ");  
 double pret = scanner.nextInt();  
  
 Sportivi.add(new Basketball(cod, nume, prenume, gen, dataNasterii, GreutateaInKilograme, InaltimeaInCentimentri,  
 StudiiSuperioare, echipa, divizie, pozitie, liga, Marcari2Puncte, Marcari3Puncte, pret));  
 }  
 }catch(java.util.InputMismatchException e)  
 {  
 scanner.next();  
 *out*.println("Tipul de date este introdus gresit");  
 }  
 }  
 //afisarea datelor citite  
  
 public static void Afisare(ArrayList<Sportivi> Sportivi)  
 {  
 for(int i = 0; i < Sportivi.size(); i++)  
 {  
 Sportivi.get(i).afisare();  
 }  
 }  
}

# **Clasa TenisAfisare Citire**

import java.util.ArrayList;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
  
/\*  
  
clasa folosita pentru citirea si afisarea datelor de la tastatura  
 pentru tipul de date tenist  
\*/  
  
//citirea tipului de date Tennis care vor fi introdusi in array-ul dinamic sportivi unde se vor afla si alte tipuri de date  
  
class TennisAfisareCitire  
{  
 public static void CitireaTenis(ArrayList<Sportivi> Sportivi, Scanner scanner, int n) throws InputMismatchException {  
 try {  
 for (int i = 0; i < n; i++) {  
 *out*.print("cod: ");  
 int cod = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Nume: ");  
 String nume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Prenume: ");  
 String prenume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("gen: ");  
 String gen = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Inaltimea: ");  
 double InaltimeaInCentimentri = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Masa: ");  
 double GreutateaInKilograme = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Data Nasterii: MM/DD/YY");  
 String dataNasterii = scanner.next();  
  
 *out*.println("\n");  
  
  
 *out*.print("Studii Superioare: ");  
 boolean StudiiSuperioare = scanner.hasNext();  
  
 *out*.print("rank: ");  
 int rank = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("divizie: ");  
 String divizie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Victorii: : ");  
 int victorii = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Infrangeri: ");  
 int infrangeri = scanner.nextInt();  
  
 *out*.println("\n");  
  
 Sportivi.add(new Tenis(cod, nume, prenume, gen, dataNasterii, GreutateaInKilograme, InaltimeaInCentimentri,  
 StudiiSuperioare, rank, divizie, victorii, infrangeri));  
 }  
 }catch(java.util.InputMismatchException e)  
 {  
 scanner.next();  
 *out*.println("Tipul de date este introdus gresit");  
 }  
 }  
//afisarea datelor citite  
  
 public static void Afisare(ArrayList<Sportivi> Sportivi)  
 {  
 for(int i = 0; i < Sportivi.size(); i++)  
 {  
 Sportivi.get(i).afisare();  
 }  
 }  
}

# **Clasa Inot Afisare Citire**

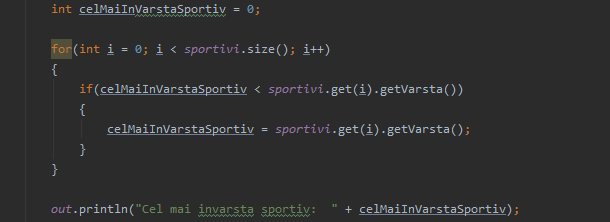
import java.util.ArrayList;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
  
/\*  
  
clasa folosita pentru citirea si afisarea datelor de la tastatura  
 pentru tipul de date inotator  
\*/  
  
//citirea tipului de dateinotatori care vor fi introdusi in array-ul dinamic sportivi unde se vor afla si alte tipuri de date  
class InotAfisareCitire  
{  
 public static void CitireaInotatori(ArrayList<Sportivi> Sportivi, Scanner scanner, int n) throws InputMismatchException {  
 try {  
 for (int i = 0; i < n; i++) {  
 *out*.print("cod: ");  
 int cod = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("Nume: ");  
 String nume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Prenume: ");  
 String prenume = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("gen: ");  
 String gen = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Inaltimea: ");  
 double InaltimeaInCentimentri = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Masa: ");  
 double GreutateaInKilograme = scanner.nextDouble();  
  
 *out*.println("\n");  
  
 *out*.print("Data Nasterii: MM/DD/YY");  
 String dataNasterii = scanner.next();  
  
 *out*.println("\n");  
  
  
 *out*.print("Studii Superioare: ");  
 boolean StudiiSuperioare = scanner.hasNext();  
  
 *out*.print("rank: ");  
 int rank = scanner.nextInt();  
  
 *out*.println("\n");  
  
 *out*.print("divizie: ");  
 String divizie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Categorie: ");  
 String Categorie = scanner.next();  
  
 *out*.println("\n");  
  
 *out*.print("Nationala: ");  
 String Nationala = scanner.next();  
  
 *out*.println("\n");  
  
 Sportivi.add(new Inot(cod, nume, prenume, gen, dataNasterii, GreutateaInKilograme, InaltimeaInCentimentri,  
 StudiiSuperioare, rank, divizie, Categorie, Nationala));  
 }  
 }catch(java.util.InputMismatchException e)  
 {  
 scanner.next();  
 *out*.println("Tipul de date este introdus gresit");  
 }  
 }  
  
//afisarea datelor citite  
  
 public static void Afisare(ArrayList<Sportivi> Sportivi)  
 {  
 for(int i = 0; i < Sportivi.size(); i++)  
 {  
 Sportivi.get(i).afisare();  
 }  
 }  
}

# **Clasa Test**

import java.time.\*;  
import java.util.ArrayList;  
import java.util.Scanner;  
import static java.lang.System.\*;  
  
//clasa pentru testarea solutiilor pentru exercitiile propuse fara a introduce date de la tastatura  
  
public class Test  
{  
  
 public static ArrayList<Sportivi> *sportivi* = new ArrayList<Sportivi>(); //Array-uL dinamic global care va fi folosit pentru prelucrarea datelor si condiitlor  
 public static Scanner *sc* = new Scanner(System.*in*); //Scanner global care va fi acceptat ca argument in fiecare metoda pentru a evita redeclararea  
  
 public static void main(String[] args)  
 {  
  
 //Sportivi creati manual pentur testarea si indeplinirea fiecare dintre codintii  
  
  
 Sportivi fotbalist1 = new Fotbal(1, "Ion", "Vasile", "Masculin","10/02/2002",  
 62.2, 1.80, false, "Bayern Munich", "Bundesliga", "Mijlocas-Central", 13, 0, 4421412);  
 Sportivi fotbalist2 = new Fotbal(2, "Bogdan", "Nurmagomedov", "Masculin","10/12/2002",  
 80, 1.60, false, "Real Madrid", "LaLiga", "Atacant-Central", 50, 3, 421412);  
  
 Sportivi basketbalist1 = new Basketball(1, "Dan", "Bryant", "Masculin", "10/02/2008", 78.9, 1.96, false, "Chicago Bulls", "Liga Nord Americana",  
 "NBL", "Dunker", 30, 63, 5215125);  
 Sportivi basketbalist2 = new Basketball(3, "Dan", "Bryant", "Masculin","05/10/1972", 80.5, 2.03, true, "Miami Heat", "Liga Nord Americana",  
 "NBL", "Defender", 55, 123, 231215125);  
  
 Sportivi Tenisista1 = new Tenis(4, "Valeria", "Cuteleaba", "Femenin","05/01/2010", 45.5, 1.60, false, 103,  
 "Juniori", 10, 1);  
  
 Sportivi Tenisista2 = new Tenis(5, "Daniela", "ManaScurta", "Femenin","05/01/2012", 35.5, 1.56, false, 103,  
 "Juniori", 5, 0);  
  
 Sportivi Inotator1 = new Inot(6, "Valeria", "Robu", "Femenin","05/01/1990", 65.3, 1.72, true, 23,  
 "Divizia1", "Seniori", "Moldova");  
  
 Sportivi Inotator2 = new Inot(7, "Inna", "Salimer", "Femenin","05/01/2000", 85.3, 1.72, false, 23,  
 "Divizia3", "Majori", "Romania");  
  
 //adaugarea fiecarui sportiv de diferit tip in array-ul dinamic  
  
 *sportivi*.add(fotbalist1);  
 *sportivi*.add(fotbalist2);  
 *sportivi*.add(basketbalist1);  
 *sportivi*.add(basketbalist2);  
 *sportivi*.add(Tenisista1);  
 *sportivi*.add(Tenisista2);  
 *sportivi*.add(Inotator1);  
 *sportivi*.add(Inotator2);  
  
 //apelarea metodelor ce ne permite citirea de la tastatura  
  
 *FotbalAfisareCitire*(*sc*);  
 *BasketAfisareCitire*(*sc*);  
 *TenisAfisareCitire*(*sc*);  
 *InotAfisareCitire*(*sc*);  
  
  
 //exercitiu 1  
  
 int celMaiInVarstaSportiv = 0;  
  
 for(int i = 0; i < *sportivi*.size(); i++)  
 {  
 if(celMaiInVarstaSportiv < *sportivi*.get(i).getVarsta())  
 {  
 celMaiInVarstaSportiv = *sportivi*.get(i).getVarsta();  
 }  
 }  
  
 *out*.println("Cel mai invarsta sportiv: " + celMaiInVarstaSportiv);  
 //exercitiul 2  
  
 double medie = 0;  
  
 for(int i = 0; i < *sportivi*.size(); i++)  
 {  
 medie = *sportivi*.get(i).getGreutateInKilograme() / i;  
 }  
  
 *out*.printf("mediaGreutate: %.2f", medie);  
 *out*.println("\n");  
  
  
 //exercitiul 3  
  
 *out*.println("Lista sportivilor de gen femenin sub 12 ani care joaca Tenis: ");  
  
 for(int i = 0; i < *sportivi*.size(); i++)  
 {  
 if(*sportivi*.get(i).getProbaSportiva() == "tenis")  
 {  
 if(*sportivi*.get(i).getVarsta() <= 12)  
 {  
 *sportivi*.get(i).afisare();  
 }  
 }  
 }  
  
 //exercitiul 4  
  
 double inaltimeMax = 0;  
 String ProbaInaltimeMaX = null;  
  
 for(int i = 0; i < *sportivi*.size(); i++)  
 {  
 if(*sportivi*.get(i).getInaltimeaInCentimetri() > inaltimeMax)  
 {  
 inaltimeMax = *sportivi*.get(i).getInaltimeaInCentimetri();  
 ProbaInaltimeMaX = *sportivi*.get(i).getProbaSportiva();  
 }  
 }  
  
 *out*.println("\n");  
 *out*.print("Proba sportiva practicata de cel mai inalt sportiv este: " + ProbaInaltimeMaX);  
  
  
 //exercitiul 5  
  
 *out*.println("cea mai inalta femeie cu studii superioare ce practica inotul: ");  
  
 double ceaMaiInaltaFemeie = 0;  
  
 for(int i = 0; i < *sportivi*.size(); i++)  
 {  
 if(*sportivi*.get(i).getGen() == "Femenin")  
 {  
 if(*sportivi*.get(i).getInaltimeaInCentimetri() > ceaMaiInaltaFemeie)  
 {  
 if(*sportivi*.get(i).getStudiiSuperioare() == true)  
 {  
 if (*sportivi*.get(i).getProbaSportiva() == "inot")  
 {  
 *sportivi*.get(i).afisare();  
 }  
 }  
 }  
 }  
 }  
  
 }  
  
 //metode ce pot fi folosite pentru citirea datelor de la tastatura de utilizator si ce vor afisa obiectul  
 //metoda pentru afisarea si citirea tipului de sportiv fotbalisti  
  
 static void FotbalAfisareCitire(Scanner sc)  
 {  
 *out*.println("Cati Fotbalisti doriti sa introduceti: ");  
 int n = sc.nextInt();  
  
 FotbalistiAfisareCitire.*CitireaFotbalisti*(*sportivi*, sc, n );  
 FotbalistiAfisareCitire.*Afisare*(*sportivi*);  
 }  
  
 //metoda pentru afisarea si citirea tipului de sportiv basketballist  
  
 static void BasketAfisareCitire(Scanner sc)  
 {  
 *out*.println("Cati basketbalisti doriti sa introduceti: ");  
 int n = sc.nextInt();  
  
 BasketAfisareCitire.*CitireaBasketball*(*sportivi*, sc, n);  
 BasketAfisareCitire.*Afisare*(*sportivi*);  
 }  
  
 //metoda pentru afisarea si citirea tipului de sportiv tenisist  
  
 static void TenisAfisareCitire(Scanner sc)  
 {  
 *out*.println("Cati tenisisti doriti sa introduceti: ");  
 int n = sc.nextInt();  
  
 TennisAfisareCitire.*CitireaTenis*(*sportivi*, sc, n);  
 TennisAfisareCitire.*Afisare*(*sportivi*);  
 }  
  
 //metoda pentru afisarea si citirea tipului de sportiv inotator  
  
 static void InotAfisareCitire(Scanner sc)  
 {  
 *out*.println("Cati Inotatori doriti sa introduceti: ");  
 int n = sc.nextInt();  
  
 InotAfisareCitire.*CitireaInotatori*(*sportivi*, sc , n);  
 InotAfisareCitire.*Afisare*(*sportivi*);  
  
 }  
}

# 1.Afișarea celui mai în vârstă sportiv care practică fotbal

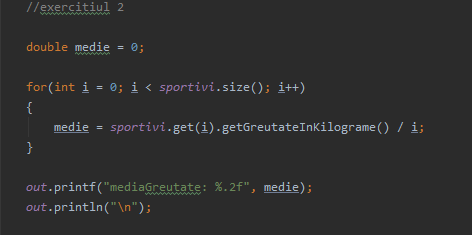
**Cod**



**Rezultat**   


2. Afișarea greutății medii a tuturor sportivilor

**Cod**

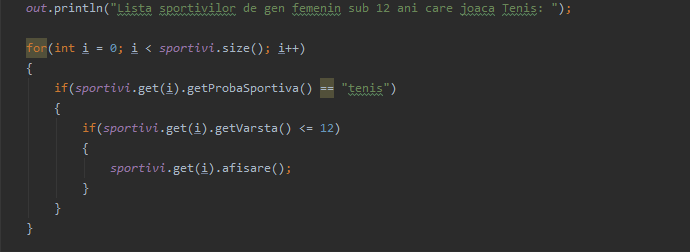


**Rezultat**

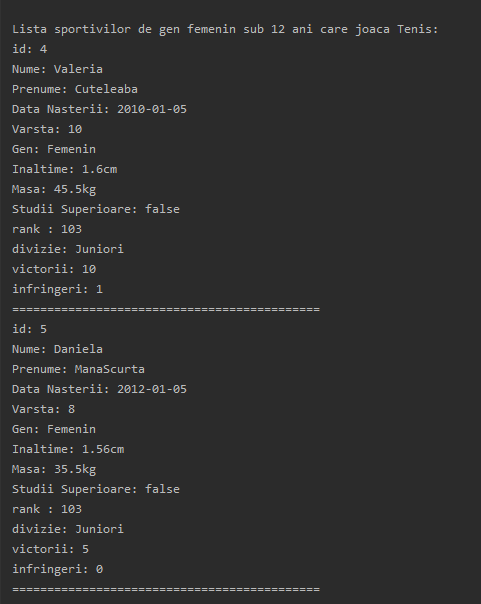


3. Afișarea listei sportivilor de gen feminin, sub 12 ani care practică tenisul

**Cod**

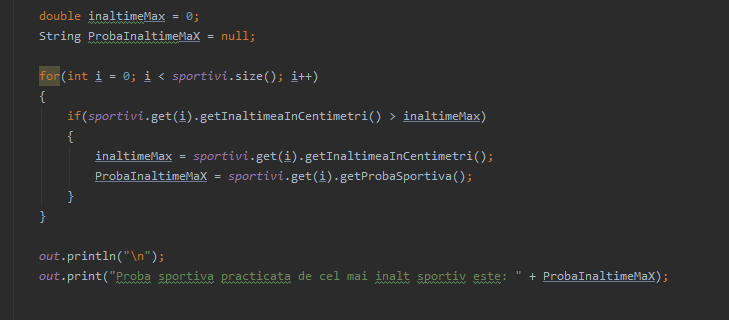


**Rezultat**



4. Afișarea denumirii probei practicată de cel mai înalt sportiv

**Cod**

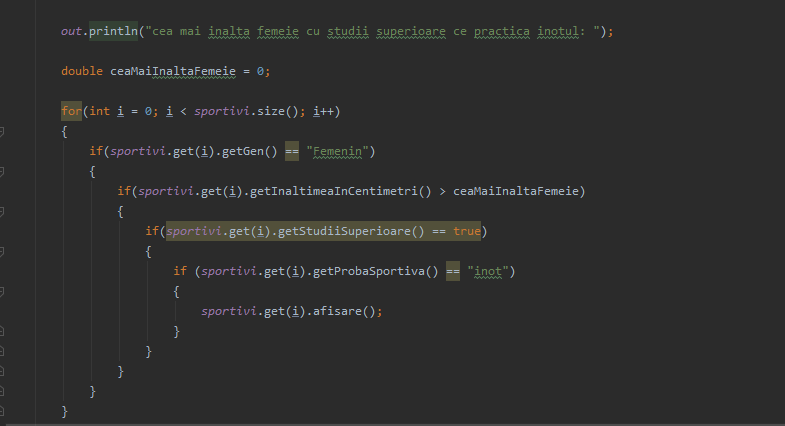


**Rezultat**



5. Afișarea denumirii probei practicată de cel mai înalt sportiv

**Cod**



**Rezultat**

