# 6-Java – Tema 3

1. Declarati si initializati 2 numere de tip int – numar1 si numar 2. Scrieți un program care afișează numerele prime între numar1 și numar2. Afiști, de asemenea, câte numere prime au fost găsite intre numar1 si numar2.
2. Avand un array cu n elemente, afișați elementul maxim din array.

public class MethEx1Ex2 {  
 public static boolean isPrime(int number) {  
  
 for (int j = 2; j < number/2 ; j++) {  
 if (number % j == 0) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
  
 public static int maxFromArray(int[] chars) {  
 int max = chars [0];  
 for (int i = 0; i < chars.length; i++) {  
 if (chars[i] > max) {  
 max = chars[i];  
 }  
 }  
 return max;  
 }  
  
}

public class MainEx1Ex2 {  
 public static void main(String[] args) {  
 //Exercise 1  
 int nr1 = 14;  
 int nr2 = 25;  
 int count = 0;  
  
 for (int i = nr1; i <= nr2; i++) {  
 if (MethEx1Ex2.*isPrime*(i) == true) {  
 System.*out*.println(i);  
 count++;  
 }  
 }  
 System.*out*.println("In intervalul dat s-au gasit " + count + " numere prime");

//Exercise 2  
 int[] array = {5,15,0,89,33,200,2,50};  
 MethEx1Ex2 myArray1 = new MethEx1Ex2();  
  
 int maxNumber = myArray1.*maxFromArray*(array);  
 System.*out*.println("Numarul maxim din sirul dat este: " + maxNumber);  
  
 }  
}

1. Realizati un joc „Ghiciti numarul” în Java, in care se va genera un număr aleatoriu și va cere utilizatorului să-l ghicească. Dacă utilizatorul ghiceste numarul, programul se va opri si se va afisa mesajul: “Ai castigat”. Dacă utilizatorul nu ghicește, programul va afișa: „Răspuns greșit, numărul tău este prea mare” sau „Răspuns greșit, numărul tău este prea mic”, in functie de comparatia dintre numarul aleatoriu generat si numarul introdus de utilizator. Programul va permite utilizatorului maxim 3 reluări după care se va opri cu mesajul „Ai pierdut”, in cazul in care nu reuseste sa ghiceasca numarul.

Sfat: Incercati sa gasiti modalitatea in care se genereaza un numar random si modul in care se citeste un numar din consola. In cazul in care nu reusiti, am sa va ofer ajutor.

import java.util.Random;  
import java.util.Scanner;  
  
public class GuessNo {  
 public static void main(String[] args) {  
 Scanner myInput = new Scanner(System.*in*);  
 System.*out*.println("Numarul random cred ca este: ");  
 int myNumber = myInput.nextInt();  
  
 // System.out.println("Numarul introdus de utilizator este: " + myNumber);  
  
 Random rand = new Random();  
 int randomNumber = rand.nextInt(10); //am dat o limita pentru a nu general numere prea mari  
 System.*out*.println("Numarul random este:" + randomNumber);  
  
 int nrIncercari = 4; //prima incercare si 3 reluari

for (int i = 0; i < nrIncercari; i++) {  
 if (myNumber == randomNumber) {  
 System.*out*.println("Ai castigat");  
 break;  
 } else if (myNumber > randomNumber) {  
 if (i == nrIncercari-1) {  
 System.*out*.println("ai pierdut");  
 } else {  
 System.*out*.println("Gresit, nr introdus e prea mare");  
 System.*out*.println("Introdu alt numar: ");  
 myNumber = myInput.nextInt();  
 }  
  
 } else if (myNumber < randomNumber) {  
 if (i == nrIncercari-1) {  
 System.*out*.println("ai pierdut");  
 } else {  
 System.*out*.println("Gresit, nr introdus e prea mic");  
 System.*out*.println("Introdu alt numar: ");  
 myNumber = myInput.nextInt();  
 }  
  
 }  
  
 }  
  
  
 }  
  
}

1. A class called Circle contains:

* Two private instance variables: radius (of type double) and color (of type String), with default value of 1.0 and “red”, respectively.
* Two overloaded constructors;
* Two public methods: getRadius() and getArea().
* Write a class which models the class Circle. Write a class TestCircle which test Circle class.

public class Circle {  
 private double radius = 1;  
 private String color = "red";  
  
 //constructor with 2 parameters  
 public Circle(double radius, String color) {  
 this.radius = radius;  
 this.color = color;  
 }  
 //constructor with 1 parameter  
 public Circle(double radius) {  
 this.radius = radius;  
 }  
  
 //contructor with no parameter  
 public Circle(){  
  
 }  
 /\*  
 public void setRadius(double radius) {  
 this.radius = radius;  
 }  
\*/  
 public double getRadius() {  
 return radius;  
 }  
  
 public static double getArea(double radius) {  
 return Math.*pow*(radius, 2) \* Math.*PI*;  
 }  
}

public class TestCircle {  
 public static void main(String[] args) {  
 Circle myCircle0 = new Circle();  
 System.*out*.println("Circle0 radius is: " + myCircle0.getRadius());  
 double myArea0 = myCircle0.*getArea*(myCircle0.getRadius());  
 System.*out*.println("Circle0 area is: " + myArea0);  
  
 Circle myCircle1 = new Circle(4, "yellow");  
 System.*out*.println("Circle1 radius is: " + myCircle1.getRadius());  
 double myArea1 = myCircle1.*getArea*(myCircle1.getRadius());  
 System.*out*.println("Circle1 area is: " + myArea1); //afisat Circle1 area folosind o variabila myArea1  
  
 Circle myCircle2 = new Circle(5);  
 System.*out*.println("Circle2 radius is: " + myCircle2.getRadius());  
 System.*out*.println("Circle2 area is: " + myCircle2.*getArea*(myCircle2.getRadius())); //afisat Circle2 area fara a retine valoarea intr-o variabila  
  
 /\*  
 Circle myCircle3 = new Circle();  
 myCircle3.setRadius(6);  
 System.out.println("Circle3 radius is: " + myCircle3.getRadius());  
 System.out.println("Circle3 area is: " + myCircle3.getArea(myCircle3.getRadius()));  
 \*/  
  
  
 }  
}

1. A class Robot contains:

* One instance variable 'x' (of type int) representing the position of robot;
* One default constructor which initialize the value to 1;
* One change(int k) method which add to the current robot x value to k (as long as k >= 1);
* One toString() method which returns the position of robot;

Write a class which models the class Robot . Write a class TestRobot which test Robot class.

public class Robot {  
 int x;  
  
 public Robot() {  
 x = 1;  
 }  
  
 public int changePos(int k) {  
 while (k >= 1)  
 return x = x + k;  
 return x;  
 }  
  
 public void getPos() {  
 System.*out*.println("Robot's position after move is: " + x);  
 }  
}

public class TestRobot {  
 public static void main(String[] args) {  
 Robot robot1 = new Robot();  
 System.*out*.println("Robot's initial position is: " + robot1.x);  
 robot1.changePos(8);  
 robot1.getPos();  
 robot1.changePos(2);  
 robot1.getPos();  
  
  
 }  
}

1. A class called Author contains:

* Three private instance variables: name (String), email (String), and gender (char of either 'm' or 'f');
* One constructor to initialize the name, email and gender with the given values;
* public Author (String name, String email, char gender) {……}
* (There is no default constructor for Author, as there are no defaults for name, email and gender.)
* public getters/setters: getName(), getEmail(), setEmail(), and getGender();
* (There are no setters for name and gender, as these attributes cannot be changed.)
* A toString() method that returns “author-name (gender) at email”, e.g., “My Name (m) at [myemail@somewhere.com](mailto:myemail@somewhere.com)”.

Write the Author class. Also write a test program called TestAuthor to test the constructor and public methods.

public class Author {  
 private String name;  
 private String email;  
 private Gender gender;  
  
 public enum Gender {  
 *m*,  
 *f* }  
  
 public Author(String name, String email, Gender gender) {  
 this.name = name;  
 this.email = email;  
 this.gender = gender;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public Gender getGender() {  
 return gender;  
 }  
  
 public static void toString(String name, Gender gender, String email) {  
 System.*out*.println(name + "(" + gender + ")" + " at " + email);  
 }  
  
}

public class TestAuthor {  
 public static void main(String[] args) {  
 Author author1 = new Author("Popescu", "popescu@gmail.com", Author.Gender.*m*);  
 System.*out*.println(author1.getName());  
 System.*out*.println(author1.getGender());  
 System.*out*.println(author1.getEmail());  
 author1.setEmail("popescu@yahoo.com");  
 System.*out*.println(author1.getEmail());  
 author1.*toString*(author1.getName(), author1.getGender(), author1.getEmail());  
  
  
  
 Author author2 = new Author ("Violeta", "vio@yahoo.com", Author.Gender.*f*);  
 String author2Name = author2.getName();  
 String author2Email = author2.getEmail();  
 Author.Gender author2Gender = author2.getGender();  
 author2.*toString*(author2Name, author2Gender, author2Email);  
  
  
 }  
}

BONUS – 1 model de problema cu care m-am confruntat recent in mediul profesional.

1. Avand la dispozitie un CSV (comma separated value – o inlantuire de valori separate intre ele prin virgula) reprezentat ca tip de data de un String (ex: 1,Ana,V,1994,ana@mail.com,Java) si un contract in care se specifica pe fiecare pozitie din CSV ce reprezinta valoarea respectiva (in cazul de fata, pe pozitia 1 se afla id-ul, pozitia a 2-a numele, 3- initiala tataului, 4- anul de nastere, 5- mail-ul, 6- limbajul de programare), scrieti o metoda care afiseaza pentru o pozitie introdusa de catre utilizator, valoarea corespunzatoare din contract insotita de valoarea din CSV. Ex: pentru pozitia 2, afiseaza -> Numele : Ana. Contruiti o alta metoda care, pentru o valoare din contract, sa afiseze valoarea din CSV. Ex: pentru mail, afiseaza -> ana@mail.com