

public class Question2 {

public static void main(String[] args) {

int [][] a = {{1,3,3,2},{4,2,5,6},{1,2,3,4},{7,8,9,4}};

if (diagonalEqualsRow(a,2))

System.out.println("Yes.");

}

public static boolean diagonalEqualsRow(int[][]myArray, int row) {

if (row >= myArray.length)

throw new IllegalArgumentException("Row out of range.");

if (myArray.length != myArray[0].length)

throw new IllegalArgumentException("Matrix not square.");

int rowSum=0;

int diagSum=0;

for (int i=0; i<myArray.length;i++) {

rowSum += myArray[row][i];

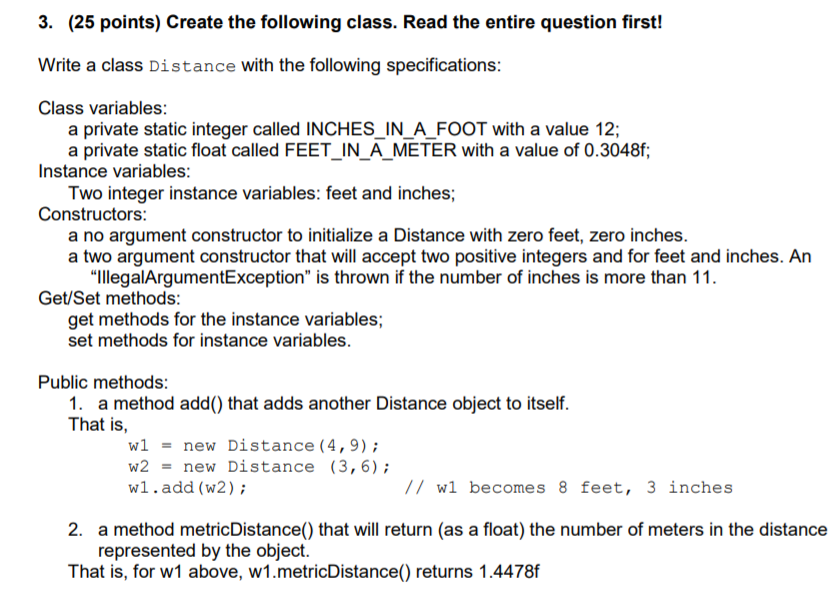
diagSum += myArray[i][i];

}

return rowSum == diagSum;

}

}



public class Distance {

/\*

\* Part a:

Write a class Distance with the following specifications:

Two instance variables: feet and inches that can be stored as integers;

a private constant called INCHES\_IN\_A\_FOOT with a value 12;

a no argument constructor to initialize a Distance with zero feet, zero inches.

Part b:

Add a two argument constructor that will accept two positive integers and for feet and inches and will initialize the instance variables. The number of feet and inches may be any integer value.

access (锟絞et锟�) methods for the instance variables;

mutator (锟絪et锟�) methods for instance variables.

Part c:

Add a method 锟絘dd锟� that accepts another Distance object as a parameter and adds the Distance of the other object to the current Distance.

That is,

w1 = new Distance(4,9);

w2 = new Distance (3,6);

w1.add(w2); // w1 becomes 8 feet, 3 inches

Part d:

Add a public method 锟組etricDistance()锟� that will return the number of meters in the distance represented by the object. This method should contain a private variable representing the number of feet in a meter (1 foot = 0.3048 meters).

\*/

private static final int INCHES\_IN\_A\_FOOT = 12;

private static final float FEET\_IN\_A\_METER = 3.28f;

private int feet, inches;

public Distance () {

feet=0;

inches=0;

}

public Distance(int f, int i) {

if (i>11)

throw new IllegalArgumentException("Inches can't be more than 11");

feet=f;

inches=i;

}

public int getInches() {

return inches;

}

public int getFeet() {

return feet;

}

public void setInches(int i) {

if (i>11)

throw new IllegalArgumentException("Inches can't be more than 11");

inches=i;

}

public void setFeet (int f) {

feet = f;

}

public void add (Distance other) {

inches = (inches + other.inches);

feet = feet + other.feet + inches/12;

inches = inches % 12;

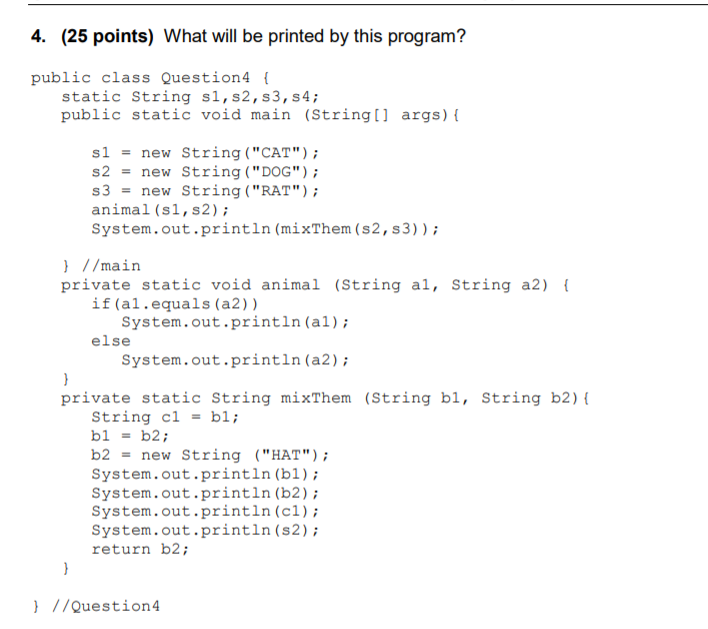
}

public float metricDistance () {

return (feet+inches/12) / FEET\_IN\_A\_METER;

}

}



public class Question4 {

static String s1,s2,s3,s4;

public static void main (String[] args){

s1 = new String("CAT");

s2 = new String("DOG");

s3 = new String("RAT");

animal(s1,s2);

System.out.println(mixThem(s2,s3));

} //main

private static void animal (String a1, String a2) {

if(a1.equals(a2))

System.out.println(a1);

else

System.out.println(a2);

}

private static String mixThem (String b1, String b2){

String c1 = b1;

b1 = b2;

b2 = new String ("HAT");

System.out.println(b1);

System.out.println(b2);

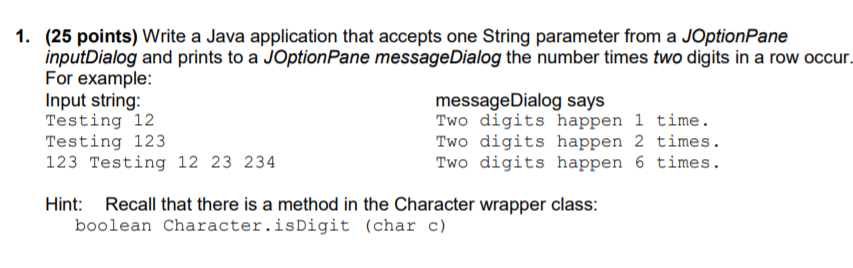
System.out.println(c1);

System.out.println(s2);

return b2;

}

} //Question4



import javax.swing.\*;

public class Question1 {

public static void main (String[] args) {

String line= JOptionPane.showInputDialog(null,"Enter a string.");

int doubleCount=0;

for (int i=0; i< line.length()-1;i++) {

if( Character.isDigit(line.charAt(i)) && Character.isDigit(line.charAt(i+1))) doubleCount++;

}

JOptionPane.showMessageDialog (null,"There are "+doubleCount + "pairs");

}

}