**Object-Oriented Programming**

Lab session #1

**Question 1**: Rectangle Visualization

Write a class Rectangle which has

* Two attributes *width* and *height* with appropriate getter methods
* A constructor that requires values for width and height of the rectangle. If either of the inputs is negative, print an error message and set the corresponding attribute to 1.
* A *visualize* method to display the rectangle using \* symbol

Write a class TestRectangle with a *main* method to instantiate 5 different rectangles and visualize them.

Answer:

public class Gettersetter{

private int width;

private int height;

public Gettersetter(int i, int j) {

}

public int getwidth() {

return width;

}

public void setwidth(int width) {

this.width = width;

if(width <0) {

System.out.print("Error");

this.width = 1;

}

}

public int getheight() {

return height;

}

public void setheight(int height) {

this.height = height;

if(height <0) {

System.out.print("Error");

this.height = 1;

} } }

import java.util.Scanner;

public class DrawRectangle {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Gettersetter input = new Gettersetter(31,23);

System.out.println("Width: ");

int width = scanner.nextInt();

input.setwidth(width);

System.out.println("Height: ");

int height = scanner.nextInt();

input.setheight(height);

for(int i = 1; i <= height; i++) {

for(int j = 1; j<= width; j++) {

if(j>=2 && j<=width-1 && i>=2 && i<=height-1) System.out.print(" ");

else System.out.print("\* ");

}

System.out.println();

}

}

}

import java.util.Scanner;

public class TestRectangle {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Gettersetter input = new Gettersetter(31,23);

System.out.println("Width: ");

int width1 = scanner.nextInt();

input.setwidth(width1);

System.out.println("Height: ");

int height1 = scanner.nextInt();

input.setheight(height1);

System.out.println("Width: ");

int width2 = scanner.nextInt();

input.setwidth(width2);

System.out.println("Height: ");

int height2 = scanner.nextInt();

input.setheight(height2);

System.out.println("Width: ");

int width3 = scanner.nextInt();

input.setwidth(width3);

System.out.println("Height: ");

int height3 = scanner.nextInt();

input.setheight(height3);

System.out.println("Width: ");

int width4 = scanner.nextInt();

input.setwidth(width4);

System.out.println("Height: ");

int height4 = scanner.nextInt();

input.setheight(height4);

System.out.println("Width: ");

int width5 = scanner.nextInt();

input.setwidth(width5);

System.out.println("Height: ");

int height5 = scanner.nextInt();

input.setheight(height5);

for(int i = 1; i <= height1; i++) {

draw\_width(width1);

System.out.println();

}

System.out.println();

for(int i = 1; i <= height2; i++) {

draw\_width(width2);

System.out.println();

}

System.out.println();

for(int i = 1; i <= height3; i++) {

draw\_width(width3);

System.out.println();

}

System.out.println();

for(int i = 1; i <= height4; i++) {

draw\_width(width4);

System.out.println();

}

System.out.println();

for(int i = 1; i <= height5; i++) {

draw\_width(width5);

System.out.println();

}

}

public static void draw\_width(int m){

for(int star = 1; star <= m; star++) {

System.out.print("\*");

}

}

}

**🡪**

**Question 2**: Triangle Verification

Write a Triangle class which has 3 attributes, the length of 3 sides. The class has appropriate constructor and get methods. This class has a method String verify() to check and return type of the Triangle. The types can be Not Triangle, Equilateral, Isosceles or Scalene. Create another class which has a main() method to receive the length of 3 sides, verify it and display the result.

Answer:

public class Trianglergetset {

private int s1;

private int s2;

private int s3;

public Trianglergetset(int i, int j, int k) {

}

public int gets1() {

return s1;

}

public void sets1(int s1) {

this.s1 = s1;

}

public int gets2() {

return s2;

}

public void sets2(int s2) {

this.s2 = s2;

}

public int gets3() {

return s3;

}

public void sets3(int s3) {

this.s3 = s3;

}

}

import java.util.Scanner;

public class Triangle {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Trianglergetset input = new Trianglergetset(100,100,100);

System.out.println("Side 1: ");

int s1 = scanner.nextInt();

input.sets1(s1);

System.out.println("Side 2: ");

int s2 = scanner.nextInt();

input.sets2(s2);

System.out.println("Side 3: ");

int s3 = scanner.nextInt();

input.sets3(s3);

if(s1 < s2+s3 || s2 < s1+s3 || s3 < s1+s2) {

System.out.println("Is Triangle");

}

if(s1 == s2 && s2 == s3) {

System.out.println("Equilateral");

}

if(s1 == s2 && s2 != s3 || s1 == s3 && s1 != s3 || s2 == s3 && s2 != s1)

System.out.println("Isosceles");

}

}

**Question 3**: Distance

Write a Point class that has private attributes for coordinates x and y. The class has constructor to get values for x and y of the point. In the class WITHOUT having getter methods for x and y, write a method

*double distance(Point target)*

to compute the distance from the current point and the given target point.

Note: the distance *d* between two points A and B can be computed with the following formula

𝑑 = √(𝑥𝐴 − 𝑥𝐵)2 + (𝑦𝐴 − 𝑦𝐵)2

Write a class with a main method to test the class Point and the distance method

double distance;

double inx = x;

double iny = y;

double z = Math.pow(5-x, 2);

double k = Math.pow(6-y, 2);

distance = Math.sqrt(z+k);

System.out.print("Distance = " +distance);

} public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Point p = new Point(5,6);

double xt = scanner.nextInt(); // input x target

double yt = scanner.nextInt(); // input y target

distance(xt, yt); } }

package OOPlab;

import java.util.Scanner;

public class Point {

private double xp;

private double yp;

public void setX(double xp) {

this.xp = xp;

}

public void setY(double yp) {

this.yp = yp;

}

public Point(double xp, double yp) {}

public static void distance(double x, double y) {

**Question 4**: E-commerce Order

You are required to program Order and Item class satisfied the requirements as follows:

1. Each order has an ID
2. Each order has a list of Items
3. Each item has an ID, a name and a price
4. Each class has appropriate constructors, get and set methods
5. Class Order has a method double calculateAverageCost() to calculate the average of the cost of all items in an order
6. Create a class which has a main() method to receive the inputs of items of an order from keyboard and display the averageCost

Answer:

private String name;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

private double price;

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public double getSum() {

return sum;

}

public void setSum(double sum) {

this.sum = sum;

}

private double sum;

}

public class GetsetOrder {

private int n;

public int getN() {

return n;

}

public void setN(int n) {

this.n = n;

}

private int idorder;

public int getIdorder() {

return idorder;

}

public void setIdorder(int idorder) {

this.idorder = idorder;

}

private int iditem;

public int getIditem() {

return iditem;

}

public void setIditem(int iditem) {

this.iditem = iditem;

}

import java.util.Scanner;

public class Order {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

GetsetOrder input = new GetsetOrder();

int sum=0;

System.out.print("ID of an Order: ");

int idorder = scanner.nextInt();

input.setIdorder(idorder);

System.out.print("The number of Items: ");

int n = scanner.nextInt();

input.setN(n);

for( int i = 1; i <= n; i++) {

inputitem();

} for( int i = 1; i<=n; i++) {

System.out.print("Price of Item " +i+ ":" );

double price = scanner.nextDouble();

input.setPrice(price);

sum +=price;

}

sum +=price;

} calculateAverageCost(sum, n);

System.out.println("Averagecost: " +calculateAverageCost(sum, n));

} public static void inputitem() {

Scanner scanner = new Scanner(System.in);

GetsetOrder input = new GetsetOrder();

System.out.print("The name of Item: ");

String name = scanner.nextLine();

input.setName(name);

System.out.print("ID of item:");

int iditem = scanner.nextInt();

input.setIditem(iditem);

}

public static double calculateAverageCost(double sum, int n) {

GetsetOrder input = new GetsetOrder();

double average;

return average = sum/n;

} }

=>