**"Angel Kanchev" University of Ruse**

**Coursework**

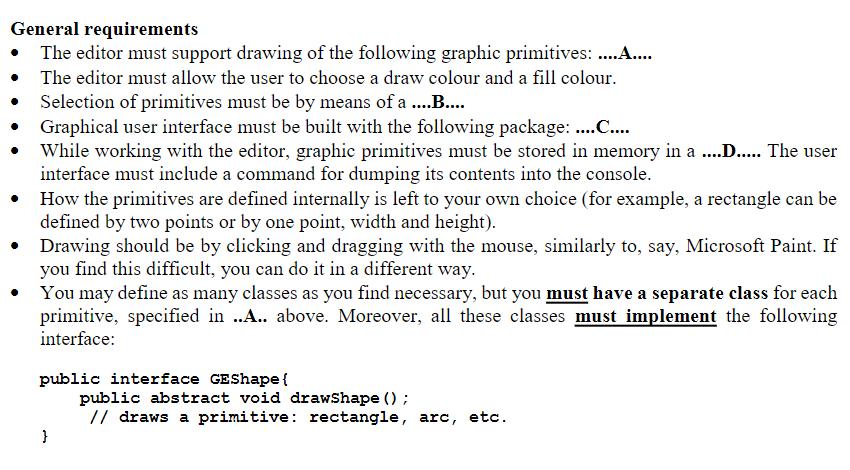
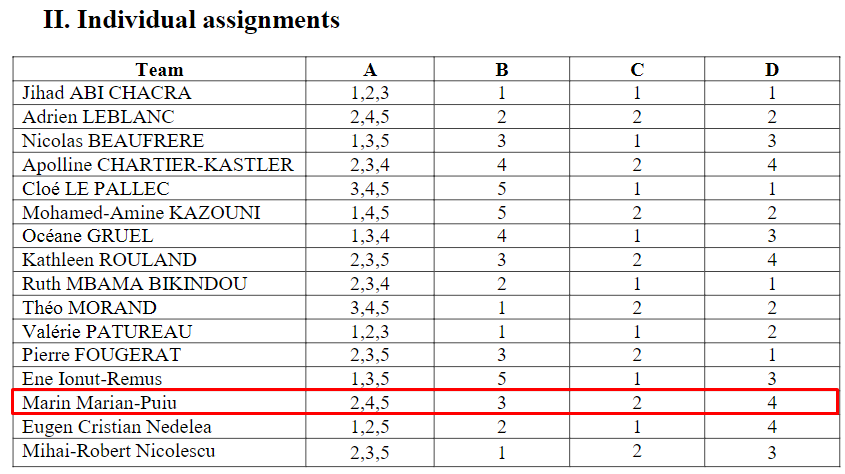
**(Programming Languages)**

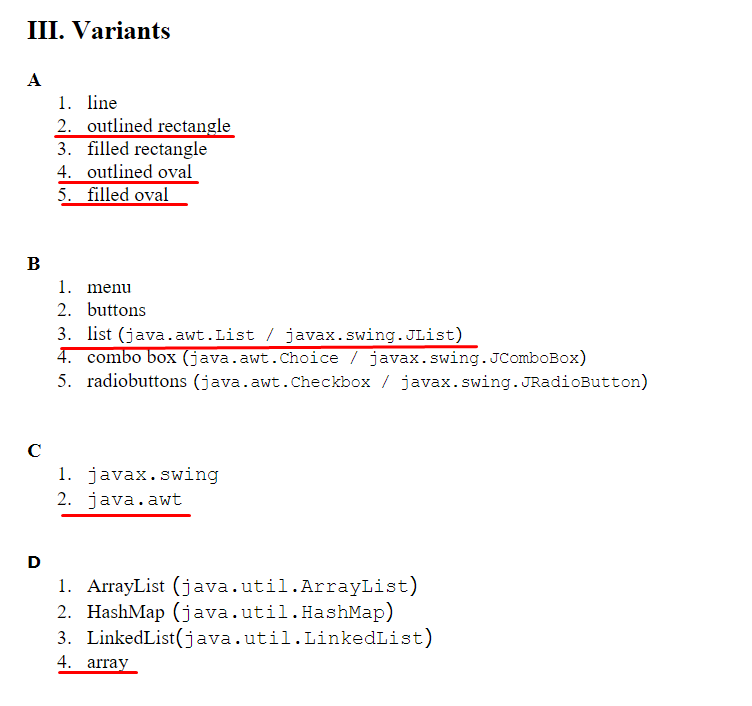
**Students: Marin Marian Puiu**

**Nicolescu Mihai-Robert**

**Group: Erasmus+**

**Lecturers: Georgie Georgiev**





**Source code:**

**draw.java**

package assignment;

import java.awt.Button;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Frame;

import java.awt.Graphics;

import java.awt.Panel;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionAdapter;

import java.awt.event.MouseMotionListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import javax.swing.JColorChooser;

public class draw extends Frame implements ActionListener{

private static final long serialVersionUID = 1L;

Button b1; //color selector

//selection of primitives

Button b2; //outlined rectangle

Button b3; //outlined oval

Button b4; //filled oval

Button b5; //this dumps the shapes to the console

int x1, y1, x2, y2;

int n = 0;

Panel p;

String currentButton;

Color newColor; //this parameter changes when the color is changed using the color picker

GEShape shape;

GEShape arr[];

java.awt.List l;

public draw() {

p = new Panel();

setSize(800,600);

p.setBounds(50,50,200,150);

p.setBackground(Color.red);

setBackground(Color.white);

setLayout(new FlowLayout());

p.setLayout(new FlowLayout());

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {

System.exit(0);

}});

addMouseListener(mouseHandler);

addMouseMotionListener(mouseMotionHandler);

l= new java.awt.List(5);

l.add("Color selector");

l.add("Outlined rectangle");

l.add("Outlined oval");

l.add("Filled oval");

l.add("Dump shapes to console");

add(l);

currentButton = "b2"; //the default selected shape is the rectangle

newColor = Color.black;

l.addActionListener(this);

arr = new GEShape[n];

setVisible(true);

}

public static GEShape[] addX(int n, GEShape arr[], GEShape x)

{

List<GEShape> arrlist = new ArrayList<GEShape>(Arrays.asList(arr));

arrlist.add(x);

arr = arrlist.toArray(arr);

return arr;

}

public void actionPerformed(ActionEvent e) {

if (l.getSelectedIndex() == 0) {

//here we don't set the currentbutton to b1 to preserve the currently selected shape

newColor = JColorChooser.showDialog(null, "Choose a color", Color.RED);

}

if (l.getSelectedIndex() == 1) {

currentButton = "b2";

}

if (l.getSelectedIndex() == 2) {

currentButton = "b3";

}

if (l.getSelectedIndex() == 3) {

currentButton = "b4";

}

if (l.getSelectedIndex() == 4) {

System.out.println("outputting the shapes:");

for (GEShape i: arr) {

System.out.println(i);

}

}

}

public MouseListener mouseHandler = new MouseAdapter() {

@Override

public void mousePressed(MouseEvent e) {

x1 = x2 = e.getX();

y1 = y2 = e.getY();

repaint();

}

@Override

public void mouseReleased(MouseEvent e) {

x2 = e.getX();

y2 = e.getY();

repaint();

arr = addX(n,arr,shape);

}

};

public MouseMotionListener mouseMotionHandler = new MouseMotionAdapter() {

@Override

public void mouseDragged(MouseEvent e) {

x2 = e.getX();

y2 = e.getY();

repaint();

}

};

@Override

public void paint(Graphics g) {

super.paint(g);

g.setColor(newColor);

shape = null;

if(currentButton == "b2") {

outlinedRectangle r = new outlinedRectangle();

r.drawShape(g, x1, x2, y1, y2);

shape = r;

}

if(currentButton == "b3") {

outlinedOval o = new outlinedOval();

o.drawShape(g, x1, x2, y1, y2);

shape = o;

}

if(currentButton == "b4") {

filledOval o = new filledOval();

o.drawShape(g, x1, x2, y1, y2);

shape = o;

}

}

public static void main(String[] args) {

new draw();

}

}

**FilledOval.java**

package assignment;

import java.awt.Graphics;

public class filledOval implements GEShape{

@Override

public void drawShape(Graphics g, int x1, int x2, int y1, int y2) {

int x = Math.min(x1, x2);

int y = Math.min(y1,y2);

int width = Math.abs(x1-x2);

int height = Math.abs(y1-y2);

g.fillOval(x,y,width,height);

}

}

**GEShape.java**

package assignment;

import java.awt.Graphics;

public interface GEShape {

public abstract void drawShape(Graphics g, int x1, int x2, int y1, int y2);

}

**OutlinedOval.java**

package assignment;

import java.awt.Graphics;

public class outlinedOval implements GEShape {

@Override

public void drawShape(Graphics g, int x1, int x2, int y1, int y2) {

int x = Math.min(x1, x2);

int y = Math.min(y1,y2);

int width = Math.abs(x1-x2);

int height = Math.abs(y1-y2);

g.drawOval(x,y,width,height);

}

}

**OutlinedRectangle.java**

package assignment;

import java.awt.Graphics;

public class outlinedRectangle implements GEShape {

public void drawShape(Graphics g, int x1, int x2, int y1, int y2) {

int x = Math.min(x1, x2);

int y = Math.min(y1,y2);

int width = Math.abs(x1-x2);

int height = Math.abs(y1-y2);

g.drawRect(x,y,width,height);

}

}