객체지향 프로그래밍 및 실습(Object Oriented Programming with Java)

과제2(HomeWork2)

컴퓨터학부 다반 20132431 이진우(Lee Jin Woo)

Source Code (MyRobot.java, RobotViewerClass.java, RobotClass.java, BarClass.java, CenterClass.java)

**- MyRobot.java**

**import** java.awt.Graphics2D;

**public** **interface** MyRobot

{

**public** **void** draw(Graphics2D g2);

**public** **void** move();

}

**- RobotClass.java**

**import** javax.swing.JPanel;

**import** javax.swing.JRadioButton;

**import** javax.swing.JButton;

**import** javax.swing.ButtonGroup;

**import** java.awt.BorderLayout;

**import** java.awt.FlowLayout;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**public** **class** RobotClass **extends** JPanel

{

**private** JPanel top;

**private** JButton button;

**private** JRadioButton redBtn;

**private** JRadioButton greenBtn;

**private** JRadioButton blueBtn;

**private** JRadioButton orangeBtn;

**private** JRadioButton yellowBtn;

**private** JRadioButton blackBtn;

**private** ButtonGroup group;

**private** String CurrentStateColor;

**private** String CurrentMoving;

**private** BarClass barClass;

**private** RadioBtnListener RBListen;

**private** ButtonListener BTListen;

**public** RobotClass()

{

top=**new** JPanel();

redBtn=**new** JRadioButton("Red");

greenBtn=**new** JRadioButton("Green");

blueBtn=**new** JRadioButton("Blue");

orangeBtn=**new** JRadioButton("Orange");

yellowBtn=**new** JRadioButton("Yellow");

blackBtn=**new** JRadioButton("Black");

group=**new** ButtonGroup();

button=**new** JButton("Move my Robot");

barClass=**new** BarClass();

CurrentStateColor="Red";

CurrentMoving="Stop";

RBListen=**new** RadioBtnListener();

BTListen=**new** ButtonListener();

button.addActionListener(BTListen);

**this**.setLayout(**new** BorderLayout());

top.setLayout(**new** FlowLayout());

group.add(redBtn);

group.add(greenBtn);

group.add(blueBtn);

group.add(orangeBtn);

group.add(yellowBtn);

group.add(blackBtn);

redBtn.addActionListener(RBListen);

greenBtn.addActionListener(RBListen);

blueBtn.addActionListener(RBListen);

orangeBtn.addActionListener(RBListen);

yellowBtn.addActionListener(RBListen);

blackBtn.addActionListener(RBListen);

top.add(redBtn);

top.add(greenBtn);

top.add(blueBtn);

top.add(orangeBtn);

top.add(yellowBtn);

top.add(blackBtn);

redBtn.setSelected(**true**);

**this**.add(barClass,BorderLayout.*CENTER*);

**this**.add(button,BorderLayout.*SOUTH*);

**this**.add(top,BorderLayout.*NORTH*);

}

**class** RadioBtnListener **implements** ActionListener

{

**public** **void** actionPerformed(ActionEvent event)

{

**if** (redBtn.isSelected())

CurrentStateColor="Red";

**else** **if** (greenBtn.isSelected())

CurrentStateColor="Green";

**else** **if** (blueBtn.isSelected())

CurrentStateColor="Blue";

**else** **if** (orangeBtn.isSelected())

CurrentStateColor="Orange";

**else** **if** (yellowBtn.isSelected())

CurrentStateColor="Yellow";

**else**

CurrentStateColor="Black";

barClass.receivedColor(CurrentStateColor);

}

}

**class** ButtonListener **implements** ActionListener

{

**public** **void** actionPerformed(ActionEvent event)

{

**if** (CurrentMoving.equals("Stop"))

CurrentMoving="Move";

**else**

CurrentMoving="Stop";

barClass.receivedNotice(CurrentMoving);

}

}

}

**- BarClass.java**

**import** javax.swing.JPanel;

**import** javax.swing.JComboBox;

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**public** **class** BarClass **extends** JPanel

{

**private** JComboBox box;

**private** String CurrentState;

**private** ComboBoxListener CBListen;

**private** CenterClass centerClass;

**public** BarClass()

{

box=**new** JComboBox();

CBListen=**new** ComboBoxListener();

centerClass=**new** CenterClass();

box.addItem("Rectangle");

box.addItem("Ellipse");

box.addItem("Square");

box.addItem("Circle");

box.addItem("RoundRectangle");

box.addActionListener(CBListen);

**this**.setLayout(**new** BorderLayout());

**this**.add(box,BorderLayout.*NORTH*);

**this**.add(centerClass,BorderLayout.*CENTER*);

}

**public** **void** receivedColor(String inputColor)

{

centerClass.LastColor(inputColor);

}

**public** **void** receivedNotice(String inputState)

{

centerClass.LastReceive(inputState);

}

**class** ComboBoxListener **implements** ActionListener

{

**public** **void** actionPerformed(ActionEvent event)

{

CurrentState=(String)box.getSelectedItem();

centerClass.LastShape(CurrentState);

}

}

}

**- CenterClass.java**

**import** java.awt.Color;

**import** java.awt.Graphics;

**import** java.awt.Graphics2D;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**import** java.awt.geom.Ellipse2D;

**import** java.awt.geom.Rectangle2D;

**import** java.awt.geom.RectangularShape;

**import** java.awt.geom.RoundRectangle2D;

**import** java.util.ArrayList;

**import** javax.swing.JPanel;

**import** javax.swing.Timer;

**class** CenterClass **extends** JPanel **implements** MyRobot

{

**private** MyMouseListener MSListen;

**private** **double** x;

**private** **double** y;

**private** **double** x1;

**private** **double** y1;

**private** **double** width;

**private** **double** height;

**private** String CurrentState;

**private** String CurrentStateColor;

**private** String CurrentMoving;

**private** ArrayList<RectangularShape> Container;

**private** ArrayList<Color> ColorContainer;

**private** MovingTimer MVTimer;

**private** Timer MoveTimer;

**private** **boolean** isBef;

**public** CenterClass()

{

MSListen=**new** MyMouseListener();

MVTimer=**new** MovingTimer();

MoveTimer=**new** Timer(100,MVTimer);

isBef=**true**;

CurrentStateColor="Red";

CurrentMoving="Stop";

CurrentState="Rectangle";

Container=**new** ArrayList<RectangularShape>();

ColorContainer=**new** ArrayList<Color>();

**this**.addMouseListener(MSListen);

}

**public** **void** LastColor(String inputColor)

{

CurrentStateColor=inputColor;

}

**public** **void** LastReceive(String inputState)

{

**if** (inputState.equals("Move"))

{

CurrentMoving="Move";

MoveTimer.start();

}

**else**

{

CurrentMoving="Stop";

MoveTimer.stop();

}

}

**public** **void** LastShape(String inputShape)

{

CurrentState=inputShape;

isBef=**false**;

Container.remove(Container.size()-1);

ColorContainer.remove(ColorContainer.size()-1);

}

**class** MovingTimer **implements** ActionListener

{

**public** **void** actionPerformed(ActionEvent event)

{

move();

}

}

**class** MyMouseListener **implements** MouseListener

{

**public** **void** mousePressed(MouseEvent event)

{

x=event.getX();

y=event.getY();

isBef=**true**;

}

**public** **void** mouseReleased(MouseEvent event)

{

x1=event.getX();

y1=event.getY();

**if** (x>x1)

{

**double** tmp=x;

x=x1;

x1=tmp;

}

**if** (y>y1)

{

**double** tmp=y;

y=y1;

y1=tmp;

}

width=Math.*abs*(x-x1);

height=Math.*abs*(y-y1);

isBef=**true**;

repaint();

}

**public** **void** mouseClicked(MouseEvent event){}

**public** **void** mouseEntered(MouseEvent event){}

**public** **void** mouseExited(MouseEvent event){}

}

**public** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g);

Graphics2D g2=(Graphics2D)g;

printAll(g2);

**this**.draw(g2);

}

**public** **void** draw(Graphics2D g2)

{

**if** (CurrentMoving.equals("Stop"))

{

**if** (CurrentState.equals("Rectangle"))

{

Rectangle2D.Double obj=**new** Rectangle2D.Double(x,y,width,height);

Container.add(obj);

g2.setColor(ChooseColor(CurrentStateColor));

g2.fill(obj);

g2.draw(obj);

}

**else** **if** (CurrentState.equals("Ellipse"))

{

Ellipse2D.Double obj=**new** Ellipse2D.Double(x,y,width,height);

Container.add(obj);

g2.setColor(ChooseColor(CurrentStateColor));

g2.fill(obj);

g2.draw(obj);

}

**else** **if** (CurrentState.equals("Square"))

{

**double** temp;

**if** (width<height)

temp=width;

**else**

temp=height;

Rectangle2D.Double obj=**new** Rectangle2D.Double(x,y,temp,temp);

Container.add(obj);

g2.setColor(ChooseColor(CurrentStateColor));

g2.fill(obj);

g2.draw(obj);

}

**else** **if** (CurrentState.equals("Circle"))

{

**double** temp;

**if** (width<height)

temp=width;

**else**

temp=height;

Ellipse2D.Double obj=**new** Ellipse2D.Double(x,y,temp,temp);

Container.add(obj);

g2.setColor(ChooseColor(CurrentStateColor));

g2.fill(obj);

g2.draw(obj);

}

**else** **if** (CurrentState.equals("RoundRectangle"))

{

RoundRectangle2D.Double obj=**new** RoundRectangle2D.Double(x,y,width,height,20,20);

Container.add(obj);

g2.setColor(ChooseColor(CurrentStateColor));

g2.fill(obj);

g2.draw(obj);

}

**else**

{

System.out.println("Undefined Access");

}

ColorContainer.add(ChooseColor(CurrentStateColor));

**if** (!isBef)

{

Container.remove(Container.size()-1);

ColorContainer.remove(ColorContainer.size()-1);

}

}

**else**

{

printAll(g2);

}

}

**public** Color ChooseColor(String input)

{

**if** (input.equals("Red"))

**return** Color.*red*;

**else** **if** (input.equals("Green"))

**return** Color.*green*;

**else** **if** (input.equals("Blue"))

**return** Color.*blue*;

**else** **if** (input.equals("Orange"))

**return** Color.*orange*;

**else** **if** (input.equals("Yellow"))

**return** Color.*yellow*;

**else**

**return** Color.*black*;

}

**public** **void** move()

{

**for** (**int** i=0;i<Container.size();i++)

{

RectangularShape temp=Container.get(i);

**double** x=temp.getX();

x++;

**double** y=temp.getY();

**double** width=temp.getWidth();

**double** height=temp.getHeight();

temp.setFrame(x,y,width,height);

Container.set(i, temp);

repaint();

}

}

**public** **void** printAll(Graphics2D g2)

{

**for** (**int** i=0;i<Container.size();i++)

{

g2.setColor(ColorContainer.get(i));

g2.fill(Container.get(i));

g2.draw(Container.get(i));

}

}

}

**- RobotViewerClass.java**

**import** javax.swing.JFrame;

**public** **class** RobotViewerClass

{

**public** **static** **void** main(String[] args)

{

JFrame frame=**new** JFrame();

frame.setSize(550,550);

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

frame.setTitle("Object Oriented Programming with JAVA :: Homework2");

RobotClass component=**new** RobotClass();

frame.add(component);

frame.setVisible(**true**);

}

}