**package** firstGraph;

**import** java.awt.Color;

**import** java.awt.Cursor;

**import** java.awt.Dimension;

**import** java.awt.Font;

**import** java.awt.Image;

**import** java.awt.Toolkit;

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

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

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

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

**import** java.awt.event.MouseMotionListener;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** javax.imageio.ImageIO;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**import** javax.swing.JTextArea;

**import** javax.swing.SwingConstants;

**import** javax.swing.Timer;

**import** firstGraph.Variables.btnProperties;

***@SuppressWarnings***("serial")

**public** **class** **MainWindow** **extends** **JFrame** **implements** MouseListener,

MouseMotionListener, ActionListener {

**public** **int** currentSlide = 1;

**JPanel** contentPane;

**FirstSlide** firstSlideInstance = **new** FirstSlide();

**SecondSlide** secondSlideInstance = **new** SecondSlide();

**ThirdSlide** thirdSlideInstance = **new** ThirdSlide();

**FourthSlide** fourthSlideInstance = **new** FourthSlide();

**Timer** myTimer = **new** Timer(getCurrentInstance().TIMER\_SPEED,

**MainWindow**.**this**);

**int** wordCounter = 0;

**public** **MainWindow**() {

setUpFrame();

loadUpContent();

}

**private** **void** **setUpFrame**() {

setUndecorated(**true**); // Fullscreen

setExtendedState(**JFrame**.***MAXIMIZED\_BOTH***); // Fullscreen again

setSize(**new** Dimension(**Toolkit**.*getDefaultToolkit*().getScreenSize()));

setResizable(**false**); // not resizable

setVisible(**true**); // Visible

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

setBackground(**Color**.***BLACK***);

**Main**.*dimX* = getWidth();

**Main**.*dimY* = getHeight();

addMouseMotionListener(**this**);

}

**private** **void** **loadUpContent**() {

loadImage();

setUpContentPane();

setUpText();

contentPane.add(getCurrentInstance().textArea);

contentPane.add(getCurrentInstance().label);

// JButton btnNewButton = new JButton("New button");

// btnNewButton.setBounds(88, 456, 264, 25);

// btnNewButton.setAlignmentX(LEFT\_ALIGNMENT);

// contentPane.add(btnNewButton);

openFile();

refresh();

}

**private** **void** **setUpContentPane**() {

contentPane = **new** JPanel(); // Create

contentPane.setBackground(**Color**.***BLACK***); // Background

contentPane.setBorder(**null**); // border

setContentPane(contentPane); // ???

contentPane.setLayout(**null**); // sets layout

**for** (**int** **i** = 0; i < getButtonCount(); i++) {

setUpButton(i, getButn(i).posX, getButn(i).posY, getButn(i).width,

getButn(i).heigth, getButn(i).text);

}

}

**private** **void** **setUpText**() {

getCurrentInstance().textArea = **new** JTextArea(); // creates text Area

getCurrentInstance().textArea.setBounds((**Main**.*dimX* / 4), **Main**.*dimY*

- (**Main**.*dimY* / 4), **Main**.*dimX* - (**Main**.*dimX* / 2), **Main**.*dimY*

- (**Main**.*dimY* / 10));

getCurrentInstance().textArea.setWrapStyleWord(**true**);

getCurrentInstance().textArea.setText(getCurrentInstance().text);

getCurrentInstance().textArea.setOpaque(**false**); // transparent

getCurrentInstance().textArea.setLineWrap(**true**);

getCurrentInstance().textArea.setForeground(**Color**.***WHITE***); // Font colour

getCurrentInstance().textArea.setFont(**new** Font("Gabriola", **Font**.***PLAIN***,

36));

getCurrentInstance().textArea.setEditable(**false**); // in not editable

}

**private** **void** **setUpButton**(**int** id, **int** positionX, **int** positionY, **int** width,

**int** height, **String** name) {

getCurrentInstance().button[id] = **new** JButton(name);

// getCurrentInstance().btn[id].setBorder(new BevelBorder(

// BevelBorder.LOWERED, null, null, null, null));

getCurrentInstance().button[id].setBorder(**null**);

// getCurrentInstance().btn[id].setBorder(new LineBorder(Color.blue));

getCurrentInstance().button[id]

.setHorizontalAlignment(SwingConstants.***LEFT***);

getCurrentInstance().button[id].setContentAreaFilled(**false**);

getCurrentInstance().button[id].setOpaque(**false**);

getCurrentInstance().button[id].setForeground(**Color**.***WHITE***);

getCurrentInstance().button[id].setFont(**new** Font("Trebuchet MS",

**Font**.***PLAIN***, 32));

getCurrentInstance().button[id].setBounds(positionX, positionY, width,

height);

getCurrentInstance().button[id].addMouseListener(**this**);

getCurrentInstance().button[id].addMouseMotionListener(**this**);

contentPane.add(getCurrentInstance().button[id]);

}

**private** **void** **openFile**() {

**try** {

getCurrentInstance().myFile = **new** File(getFilePath());

getCurrentInstance().filRead = **new** FileReader(

getCurrentInstance().myFile);

getCurrentInstance().bufRead = **new** BufferedReader(

getCurrentInstance().filRead);

} **catch** (**IOException** **e1**) {

**System**.***err***.println("Error in accesing text file.");

}

}

**private** **void** **loadImage**() {

**try** {

getCurrentInstance().image1 = **ImageIO**.*read*(**new** File(getPicPath()));

getCurrentInstance().myImageIcon = **new** ImageIcon(

getCurrentInstance().image1.getScaledInstance(-1,

**Main**.*dimY*, **Image**.***SCALE\_SMOOTH***));

} **catch** (**IOException** **e**) {

**System**.***err***.println("Error loading picture.");

**System**.*exit*(0);

} **catch** (**NullPointerException** **e**) {

**System**.***err***.println("Error loading picture.");

**System**.*exit*(0);

}

getCurrentInstance().label = **new** JLabel();

getCurrentInstance().label.setIcon(getCurrentInstance().myImageIcon);

getCurrentInstance().label.setBounds(calcImagePos(), 0, **Main**.*dimX*,

**Main**.*dimY*);

}

**private** **int** **calcImagePos**() {

**int** **x** = (**Main**.*dimX* - getCurrentInstance().myImageIcon.getIconWidth()) / 2;

**return** x;

}

**private** **void** **refresh**() {

revalidate();

repaint();

}

**private** **void** **addLetter**() {

**char** **pom** = 1;

**try** {

pom = (**char**) getCurrentInstance().bufRead.read();

} **catch** (**IOException** **exception**) {

// **TODO** Auto-generated catch block

exception.printStackTrace();

}

getCurrentInstance().text = getCurrentInstance().text

+ **Character**.*toString*(pom);

getCurrentInstance().textArea.setText(getCurrentInstance().text);

getCurrentInstance().textArea.revalidate(); // Recalculates the text

// area.

refresh();

}

**private** **String** **getPicPath**() {

**switch** (currentSlide) {

**case** 1:

**return** firstSlideInstance.getPIC\_PATH();

**case** 2:

**return** secondSlideInstance.getPIC\_PATH();

**case** 3:

**return** thirdSlideInstance.getPIC\_PATH();

**case** 4:

**return** fourthSlideInstance.getPIC\_PATH();

**default**:

**return** "Boha";

}

}

**private** **String** **getFilePath**() {

**switch** (currentSlide) {

**case** 1:

**return** firstSlideInstance.getFILE\_PATH();

**case** 2:

**return** secondSlideInstance.getFILE\_PATH();

**case** 3:

**return** thirdSlideInstance.getFILE\_PATH();

**case** 4:

**return** fourthSlideInstance.getFILE\_PATH();

**default**:

**return** "Boha";

}

}

**private** **btnProperties** **getButn**(**int** buttonId) {

**switch** (currentSlide) {

**case** 1:

**return** firstSlideInstance.btnSetUp[buttonId];

**case** 2:

**return** secondSlideInstance.butn[buttonId];

**case** 3:

**return** thirdSlideInstance.butn[buttonId];

**case** 4:

**return** fourthSlideInstance.butn[buttonId];

**default**:

**System**.***out***.println("piss off");

**return** **null**;

}

}

**private** **int** **getButtonCount**() {

**switch** (currentSlide) {

**case** 1:

**return** firstSlideInstance.buttonCount;

**case** 2:

**return** secondSlideInstance.buttonCount;

**case** 3:

**return** thirdSlideInstance.buttonCount;

**case** 4:

**return** fourthSlideInstance.buttonCount;

**default**:

**System**.***out***.println("piss off");

**return** 0;

}

}

**private** **Variables** **getCurrentInstance**() {

**switch** (currentSlide) {

**case** 1:

**return** firstSlideInstance;

**case** 2:

**return** secondSlideInstance;

**case** 3:

**return** thirdSlideInstance;

**case** 4:

**return** fourthSlideInstance;

**default**:

**System**.***out***.println("piss off");

**return** **null**;

}

}

***@Override***

**public** **void** **actionPerformed**(**ActionEvent** arg0) {

// **TODO** Auto-generated method stub

wordCounter++;

**if** (wordCounter > 300) {

myTimer.stop();

} **else**

addLetter();

}

***@Override***

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

// **TODO** Auto-generated method stub

**if** (event.getSource() == firstSlideInstance.button[1]

|| event.getSource() == thirdSlideInstance.button[0]) {

**System**.*exit*(0);

} **else** **if** (event.getSource() == secondSlideInstance.button[0]) {

myTimer.start();

} **else** **if** (event.getSource() == firstSlideInstance.button[0]

|| event.getSource() == secondSlideInstance.button[1]) {

currentSlide++;

loadUpContent();

}

}

***@Override***

**public** **void** **mouseMoved**(**MouseEvent** event) {

// **TODO** Auto-generated method stub

**if** (event.getSource() == **this**) {

**this**.setCursor(**Cursor**.*getPredefinedCursor*(**Cursor**.***CROSSHAIR\_CURSOR***));

} **else** **if** (event.getSource() == getCurrentInstance().button[0]

|| event.getSource() == getCurrentInstance().button[1]

|| event.getSource() == getCurrentInstance().button[2]

|| event.getSource() == getCurrentInstance().button[3]) {

**this**.setCursor(**Cursor**.*getPredefinedCursor*(**Cursor**.***HAND\_CURSOR***));

}

}

***@Override***

**public** **void** **mouseDragged**(**MouseEvent** e) {

// **TODO** Auto-generated method stub

}

***@Override***

**public** **void** **mouseEntered**(**MouseEvent** e) {

// **TODO** Auto-generated method stub

}

***@Override***

**public** **void** **mouseExited**(**MouseEvent** e) {

// **TODO** Auto-generated method stub

}

***@Override***

**public** **void** **mousePressed**(**MouseEvent** e) {

// **TODO** Auto-generated method stub

}

***@Override***

**public** **void** **mouseReleased**(**MouseEvent** e) {

// **TODO** Auto-generated method stub

}

}