Курсова Работа

**Нов Български Университет**

**F65063**

**26-May-14**

Александър Кирилов Кирилов

**Home Work 1.1**

package drawpanelgrafic;

import java.awt.\*;

import javax.swing.\*;

/\*\* \* \* @author Александър Кирилов F65063 \*/

public class DrawPanelGrafic {

/\*\* \* @param args the command line arguments \*/

//Graphic class Panel;

public static class MyCanvas extends JComponent {

@Override

public void paintComponent (Graphics g) {

//Main Rectangle with blue color;

g.setColor (Color.BLUE);

g.fillRect (10,10, 400, 400);

//Second Rectangle - White color;

g.setColor (Color.WHITE);

g.fillRoundRect(30, 30, 360, 360, 40, 50);

//Third Rectangle - Red color;

g.setColor(Color.RED);

g.fillRect (71, 71, 145, 145);

//First circle Blue

g.setColor (Color.BLUE);

g.fillArc(70, 70, 290, 290, 90, -270);

//Second circle White

g.setColor (Color.WHITE);

g.fillArc (70, 70, 289, 289, 90, 90);

}

}

public static void main(String[] args) {

// TODO code application logic here

//Създаваме нов прозорец и го визиолизира ме!!!

JFrame mainFrame = new JFrame ("Graphics");

mainFrame.getContentPane ().add (new MyCanvas ());

mainFrame.setPreferredSize (new Dimension (450, 470));

mainFrame.pack();

mainFrame.setVisible(true);

}

}

**Home Work 1.2**

/\* \* To change this template, choose Tools | Templates

\* and open the template in the editor. \*/

package hm2buttonsfinnal;

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

/\*\* \* \* @author Александър Кирилов F65063 \*/

public class HM2ButtonsFinnal {

/\*\*\* @param args the command line arguments\*/

//"Panel" контейнер, който ще ни послужи за общата Форма

public static class PanelButtons extends Frame {

//Компоненти за layoutmanger-a

Panel pnlFlow1 = new Panel ();

Panel pnlGrid = new Panel ();

Panel pnlFlow2 = new Panel ();

public PanelButtons () {

//Първи ляв контейнер.

pnlFlow1.setLayout(new GridLayout (0,1));

/\*for (int i=0; i<3; i++) { pnlGrid.add(new Button ("Buttons1"+(i+1)));}

add (pnlGrid, BorderLayout.WEST);\*/

pnlFlow1.add (new Button ("Button 11"));

pnlFlow1.add (new Button ("Button 12"));

pnlFlow1.add (new Button ("Button 13"));

add(pnlFlow1, BorderLayout.WEST);

//Втори централен контайнер.

pnlGrid.setLayout(new GridLayout (2,2));

pnlGrid.add (new Button ("Button 21"));

pnlGrid.add (new Button ("Button 22"));

pnlGrid.add (new Button ("Button 23"));

pnlGrid.add (new Button ("Button 24"));

add(pnlGrid, BorderLayout.EAST);

//Трети южен контайнер или footer

pnlFlow2.setLayout(new FlowLayout ());

pnlFlow2.add (new Button ("Button 31"));

pnlFlow2.add (new Button ("Button 32"));

pnlFlow2.add (new Button ("Button 33"));

add(pnlFlow2, BorderLayout.SOUTH);

addWindowListener(new WindowAdapter() {

@Override

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

setTitle ("Homework2.1"); //Заглавието на frame-a.

setSize (230, 170); //размера на fram-a.

setVisible (true); //Правиго Видим за потребителя.

}

}

public static void main(String[] args) {

PanelButtons panelButtons = new PanelButtons ();

}

}

**Home Work 2**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package pointedtouchpanel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

/\*\*

\*

\* @author Aleksander Kirilov f65063

\*/

public class PointedTouchPanel {

/\*\*

\* @param args the command line arguments

\*/

public static class PanelButtons extends Frame {

private Button btnRed = new Button ("Red");

private Button btnBlue = new Button ("Blue");

private Button btnGreen = new Button ("Green");

private Button btnClear = new Button ("Clear");

private Button btnExit = new Button ("Exit");

private final TextField coordField = new TextField ("Coorddinates");

private TextField xCoord = new TextField();

private TextField yCoord = new TextField();

public PanelButtons () {

//Left Panel for buttons - Color, Clear and Exit

Panel leftPanel = new Panel (new GridLayout (9,1));

leftPanel.add(btnRed);

leftPanel.add(btnBlue);

leftPanel.add(btnGreen);

leftPanel.add(btnClear);

leftPanel.add(btnExit);

//Coordinates Field

leftPanel.add(coordField);

leftPanel.add(xCoord);

leftPanel.add(yCoord);

coordField.setBackground (Color.WHITE);

coordField.setEditable (false);

//Main screen with points

add(leftPanel, BorderLayout.WEST );

//Exit the program

addWindowListener (new WindowAdapter () {

@Override

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

//Getting the Coordinates and creating the points

addMouseListener (new MouseAdapter () {

@Override

public void mousePressed(MouseEvent e) {

Frame frame = (Frame) e.getSource();

Graphics g = frame.getGraphics ();

g.setColor(Color.yellow);

g.fillOval(e.getX()-4, e.getY()- 4, 22, 22);

xCoord.setText ("" + e.getX ());

yCoord.setText ("" + e.getY ());

}

});

//Exit the program

btnExit.addActionListener (new ActionListener () {

@Override

public void actionPerformed(ActionEvent ae) {

System.exit(0);

}

});

// Clearing Desktop

btnClear.addActionListener (new ActionListener () {

@Override

public void actionPerformed(ActionEvent ae) {

repaint();

xCoord.setText ("");

yCoord.setText ("");

}

});

// Действието на бутоните в смяна цвета на задният фон

btnRed.addActionListener (new ActionListener () {

@Override

public void actionPerformed(ActionEvent ae) {

setBackground (Color.RED);

xCoord.setText ("");

yCoord.setText ("");

}

});

btnBlue.addActionListener (new ActionListener () {

@Override

public void actionPerformed(ActionEvent ae) {

setBackground (Color.BLUE);

xCoord.setText ("");

yCoord.setText ("");

}

});

btnGreen.addActionListener (new ActionListener () {

@Override

public void actionPerformed(ActionEvent ae) {

setBackground (Color.GREEN);

xCoord.setText ("");

yCoord.setText ("");

}

});

//Name and Size for the Frame

setTitle ("Points");

setSize (700,350);

setVisible (true);

}

}

public static void main(String[] args) {

// TODO code application logic here

PanelButtons panelButtons = new PanelButtons();

}

}

**Home Work 3**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package boxradio;

import java.awt.\*;

import java.awt.event.\*;

/\*\*

\*

\* @author Axel

\*/

public class BoxRadio {

/\*\*

\* @param args the command line arguments

\*/

public static class ButtonCounterDemo extends Frame {

// Parts for Frame and Graphic

Panel panelLeft = new Panel();

Panel panelRight = new Panel();

Panel pnlCount = new Panel();

Panel pnlClear = new Panel();

CheckboxGroup ckbxgPlusMinus = new CheckboxGroup();

CheckboxGroup ckbxgStep = new CheckboxGroup();

Checkbox ckbxAscend = new Checkbox("Ascending",true,ckbxgPlusMinus);

Checkbox ckbxDescend = new Checkbox("Descending",false,ckbxgPlusMinus);

Checkbox ckbxStep1 = new Checkbox("Step 1",true,ckbxgStep);

Checkbox ckbxStep5 = new Checkbox("Step 5",false,ckbxgStep);

Checkbox ckbxStep10 = new Checkbox("Step 10",false,ckbxgStep);

Button btnCount = new Button("ClickMe!");

Button btnClear = new Button("Clear");

Label lblDisplay = new Label("0");

Label lblEmpty = new Label("");

int step = 1;

int counter = 0;

public ButtonCounterDemo () {

//Dobavqne na elementite na Frame

panelLeft.setLayout(new GridLayout(6,1));

panelLeft.add(ckbxAscend);

panelLeft.add(ckbxDescend);

panelLeft.add(lblEmpty);

panelLeft.add(ckbxStep1);

panelLeft.add(ckbxStep5);

panelLeft.add(ckbxStep10);

add(panelLeft,BorderLayout.WEST);

panelRight.setLayout(new GridLayout(3,1));

btnCount.setSize(60,getHeight());

btnClear.setSize(60,getHeight());

panelRight.add(btnCount);

lblDisplay.setAlignment(Label.RIGHT);

panelRight.add(lblDisplay);

panelRight.add(btnClear);

add(panelRight,BorderLayout.EAST);

// Butonni

btnCount.addActionListener(new ButtonHandler());

btnClear.addActionListener(new ButtonHandler());

//Narastvane I namalqvane na stoinostite

ckbxAscend.addItemListener(new CheckBoxHandler());

ckbxDescend.addItemListener(new CheckBoxHandler());

ckbxStep1.addItemListener(new CheckBoxHandler());

ckbxStep5.addItemListener(new CheckBoxHandler());

ckbxStep10.addItemListener(new CheckBoxHandler());

addWindowListener(new WindowClosing());

setTitle("ButtonCounterDemo");

setSize(300,200);

setResizable(false);

setVisible(true);

}

// Funkcii na butoonite

class ButtonHandler implements ActionListener {

@Override

public void actionPerformed (ActionEvent e) {

Button btn = (Button) e.getSource();

if (btn.equals(btnCount)) {

counter += step;

if ((counter < 0) || (counter > 100))

btnCount.setEnabled(false);

else

lblDisplay.setText(((Integer)(counter)).toString());

}

else

if (btn.equals(btnClear)){

lblDisplay.setText("0");

counter = 0;

btnCount.setEnabled(true);

}

}

}

//Funkciite za uvelichavane I namalqvane na stoinostite

class CheckBoxHandler implements ItemListener {

@Override

public void itemStateChanged (ItemEvent e) {

Checkbox ckb = (Checkbox)e.getSource();

if (ckb.equals(ckbxAscend)) {

if (step < 0)

step = - step;

if ((!btnCount.isEnabled())&& ((counter - step) < 0))

btnCount.setEnabled(true);

}

if (ckb.equals(ckbxDescend)) {

if (step > 0)

step = - step;

if ((!btnCount.isEnabled())&& ((counter - step) > 100))

btnCount.setEnabled(true);

}

if (ckb.equals(ckbxStep1))

step = sign(step)\*1;

else

if (ckb.equals(ckbxStep5))

step = sign(step)\*5;

else

if (ckb.equals(ckbxStep10))

step = sign(step)\*10;

}

}

//Izhod chrez Buttona X

class WindowClosing extends WindowAdapter {

@Override

public void windowClosing (WindowEvent e) {

System.exit(0);

}

}

private int sign(int s) {

if (s < 0) return -1;

else return +1;

}

}

public static void main(String[] args) {

ButtonCounterDemo btnCounter = new ButtonCounterDemo();

}

}

**Home Work 4**

package newinterfacearay2;

import java.awt.\*;

import java.awt.event.\*;

/\*\*

\*

\* @author Alex Kirilov Kirilov f65063

\*/

public class NewInterFaceAray2 extends Frame implements ActionListener {

//Elementi

Button addButton, UpButton, DownButton, LButton, RButton;

TextField inputTF;

List LList, RList;

Label LCounter, RCounter;

public NewInterFaceAray2 () {

setBackground (Color.GRAY);

setLayout (new GridBagLayout ());

GridBagConstraints p1 = new GridBagConstraints ();

//First Line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

p1.fill = GridBagConstraints.BOTH;

p1.insets = new Insets (5,5,5,5);

//TextField

p1.gridx = 0; p1.gridy = 0;

p1.gridheight = 1; p1.gridwidth = 1;

inputTF = new TextField ("",10);

add(inputTF,p1);

//Add Button

p1.gridx = 1; p1.gridy = 0;

p1.gridheight = 1; p1.gridwidth = 1;

addButton = new Button ("Add");

add(addButton,p1);

//UpButton

p1.gridx = 8; p1.gridy = 0;

p1.gridheight = 1; p1.gridwidth = 1;

UpButton = new Button ("Up");

add(UpButton,p1);

//DownButton

p1.gridx = 9; p1.gridy = 0;

p1.gridheight = 1; p1.gridwidth = 1;

DownButton = new Button ("Down");

add(DownButton,p1);

//End First Line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Second Line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Left List

p1.gridx = 0; p1.gridy = 1;

p1.gridheight = 6; p1.gridwidth = 2;

LList = new List (15);

add(LList,p1);

//Center Buttons

p1.gridx = 6; p1.gridy = 5;

p1.gridheight = 1; p1.gridwidth = 1;

LButton = new Button ("<<");

add(LButton,p1);

p1.gridx = 6; p1.gridy = 3;

p1.gridheight = 1; p1.gridwidth = 1;

RButton = new Button (">>");

add(RButton,p1);

//Right List;

p1.gridx = 8; p1.gridy = 1;

p1.gridheight = 8; p1.gridwidth = 6;

RList = new List (10);

add (RList,p1);

//End Second Line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Third Line - Counters

p1.gridx = 0; p1.gridy = 10;

LCounter = new Label ("Count : 0");

add(LCounter,p1);

p1.gridx = 9; p1.gridy = 10;

RCounter = new Label ("Count : 0");

add(RCounter,p1);

// Buttoni za otmestvane

UpButton.addActionListener ((ActionListener) this);

DownButton.addActionListener ((ActionListener) this);

inputTF.addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if(e.getKeyCode() == KeyEvent.VK\_ENTER){

if( ! inputTF.getText().trim().isEmpty()) {

LList.add(inputTF.getText().trim());

LCounter.setText("Count : " + LList.getItemCount());

inputTF.setText(null);

inputTF.requestFocus();

}

}

}

});

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if( ! inputTF.getText().trim().isEmpty()) {

LList.add(inputTF.getText().trim());

LCounter.setText("Count : " + LList.getItemCount());

inputTF.setText(null);

inputTF.requestFocus();

}

}

});

RButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(LList.getSelectedItem() != null) {

RList.add(LList.getSelectedItem());

LList.remove(LList.getSelectedItem());

LCounter.setText("Count : " + LList.getItemCount());

RCounter.setText("Count : " + RList.getItemCount());

}

}

});

RButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (RList.getSelectedItem() != null ) {

LList.add(RList.getSelectedItem());

RList.remove(RList.getSelectedItem());

LCounter.setText("Count : " + LList.getItemCount());

RCounter.setText("Count : " + RList.getItemCount());

}

}

});

addWindowListener(new WindowAdapter() {

@Override

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

pack();

setSize (500,500);

setTitle ("HomeWork");

setVisible (true);

}

public static void main(String[] args) {

NewInterFaceAray2 newInterFaceAray2 = new NewInterFaceAray2();

}

@Override

public void actionPerformed (ActionEvent e) {

if (e.getSource ().equals(UpButton)) {

if (RList.getSelectedIndex () > 0 )

{

int indexx = RList.getSelectedIndex ();

String replaced = RList.getItem (indexx-1);

RList.replaceItem(RList.getSelectedItem(), indexx-1);

RList.replaceItem(replaced, indexx);

}

}

if(e.getSource().equals(DownButton)) {

if(RList.getSelectedIndex() < RList.getItemCount() -1

&& RList.getSelectedIndex() > -1) {

int ind = RList.getSelectedIndex();

String replaced = RList.getItem(ind+1);

RList.replaceItem(RList.getSelectedItem(), ind+1);

RList.replaceItem(replaced, ind);

}

}

}

}

**Home Work 5**

package notepadhm6;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Alexander Kirilov F65053

\*/

public class NotePadHM6 extends Frame {

//Elementi

MenuBar menu;

Menu file, edit, help;

MenuItem mFNew, mFOpen, mFSave, mFExit; //MenuItem for File Menu

MenuItem mECopy, mEPaste, mECut, mEClear; // MenuItems for Edit Menu

CheckboxMenuItem mEBold, mEItalic; //CheckBox for Edit Menu

MenuItem mHAbout; // MenuItem for Help Menu;

TextArea TextPad = new TextArea ("");

String replaced = null; // For selected text;

String str = TextPad.getText();

Boolean bool = false;

Panel screen1 = new Panel ();

//For Dialog Window - Save - Open

FileDialog fd1;

Label lab1;

public NotePadHM6 ()

{

//Adding Menu Bar

menu = new MenuBar ();

setMenuBar (menu);

//Adding Menu - File

file = new Menu ("File", true);

menu.add(file);

//Menu Item - File

mFNew = new MenuItem ("New", new MenuShortcut (KeyEvent.VK\_N));

mFOpen = new MenuItem ("Open", new MenuShortcut (KeyEvent.VK\_O));

mFSave = new MenuItem ("Save", new MenuShortcut (KeyEvent.VK\_S));

mFExit = new MenuItem ("Exit", new MenuShortcut (KeyEvent.VK\_Q));

mFNew.addActionListener (new MenuListener ());

mFOpen.addActionListener (new MenuListener ());

mFSave.addActionListener (new MenuListener ());

mFExit.addActionListener (new MenuListener ());

file.add(mFNew);

file.add(mFOpen);

file.add(mFSave);

file.add(mFExit);

//Adding Menu - Edit

edit = new Menu ("Edit", true);

menu.add(edit);

//Menu Items - Edit

mECut = new MenuItem ("Cut", new MenuShortcut (KeyEvent.VK\_X));

mECopy = new MenuItem ("Copy", new MenuShortcut (KeyEvent.VK\_C));

mEPaste = new MenuItem ("Paste", new MenuShortcut (KeyEvent.VK\_V));

mEClear = new MenuItem ("Clear");

mEBold = new CheckboxMenuItem ("Bold", false);

mEItalic = new CheckboxMenuItem ("Italic", false);

mECut.addActionListener (new MenuListener ());

mECopy.addActionListener (new MenuListener ());

mEPaste.addActionListener (new MenuListener ());

mEClear.addActionListener (new MenuListener ());

mEBold.addActionListener (new MenuListener ());

mEItalic.addActionListener (new MenuListener ());

edit.add(mECut);

edit.add(mECopy);

edit.add(mEPaste);

edit.add(mEClear);

edit.add(mEBold);

edit.add(mEItalic);

//Adding Menu - About

help = new Menu ("Help", true);

menu.add(help);

//Adding Menu Item - Help

mHAbout = new MenuItem ("About", new MenuShortcut (KeyEvent.VK\_H));

help.add(mHAbout);

//Adding TextArea

add(TextPad);

addWindowListener ( new WindowAdapter (){

public void windowClosing (WindowEvent we){

System.exit(0);

}

});

pack();

setLocationRelativeTo(null);

setSize (600,400);

setTitle ("Notepad +");

setVisible (true);

}

public static void main(String[] args) {

NotePadHM6 notepad = new NotePadHM6 ();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Functions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Exit, Close Button, New , Clear

class MenuListener implements ActionListener {

//mFExit.addActionListener ( new ActionListener (){

@SuppressWarnings("null")

@Override

public void actionPerformed (ActionEvent ae) {

MenuItem menuitem = (MenuItem) ae.getSource();

if (menuitem.equals(mFExit)){

if (TextPad.getText().length() > 1 && bool == false) {

final Dialog dia = new Dialog(new Frame(), "Save ?", true);

Button yes = new Button("Yes");

Button no = new Button("No");

yes.addActionListener((ActionEvent e) -> {

dia.setVisible(false);

ActionEvent ae1 = new ActionEvent((MenuItem)mFSave, ActionEvent.ACTION\_PERFORMED, "");

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(ae1);

});

no.addActionListener((ActionEvent e) -> {

System.exit(0);

});

dia.add("South", no);

dia.add("North", yes);

dia.pack();

Dimension dd = dia.getSize();

Dimension pd = TextPad.getParent().getSize();

Point pl = TextPad.getParent().getLocation();

dia.setLocation(

pl.x + ((int) (pd.getWidth() - dd.getWidth())) / 2,

pl.y + ((int) (pd.getHeight() - dd.getHeight())) / 2

);

dia.setVisible(true);

bool = true;

} else {

System.exit(0);

}

}

//mFNew.addActionListener( new ActionListener (){

if (menuitem.equals(mFNew)){

TextPad.setText ("");

}

//mEClear.addActionListener ( new ActionListener () {

if (menuitem.equals (mEClear)){

TextPad.removeTextListener((TextListener) ae);

}

//mFOpen.addActionListener (new ActionListener () {

if (menuitem.equals(mFOpen)){

FileDialog dialo = new FileDialog(new Frame(), "Open");

dialo.setFile("\*.txt");

dialo.setVisible(true);

String sour = dialo.getDirectory() + dialo.getFile();

try {

FileReader rea = new FileReader(sour);

char[] buff = new char[4096];

int len;

TextPad.setText(" ");

while ((len = rea.read(buff)) != -1) {

String s = new String(buff, 0, len);

TextPad.setText(s);

}

if (rea != null) {

rea.close();

}

} catch (FileNotFoundException ex) {

Logger.getLogger(NotePadHM6.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(NotePadHM6.class.getName()).log(Level.SEVERE, null, ex);

}

}

//mFSave.addActionListener (new ActionListener () {

if (menuitem.equals(mFSave)) {

if (TextPad.getText().length() < 5) {

final Dialog di = new Dialog(new Frame(), "Nqma Informaciq koqto da bude zapazena", true);

// true for modal

di.add("Center", new Label("Empty File"));

di.setSize(300,200);

di.setLocationRelativeTo(null);

Button ok = new Button("Continue");

ok.addActionListener((ActionEvent e) -> {

di.setVisible(false);

});

di.add("South", ok);

di.pack();

di.setVisible(true);

di.setLocationRelativeTo(null);

} else {

mFSave.setEnabled(true);

FileDialog savefile = new FileDialog(new Frame(), "Save File", FileDialog.SAVE);

savefile.setVisible(true);

final String fullpath = savefile.getDirectory() + savefile.getFile();

File sfi = new File(fullpath);

// Now write to the file

try {

PrintWriter output;

output = new PrintWriter(new FileWriter(sfi));

output.println(TextPad.getText());

output.close();

bool = true;

} catch (IOException ex) {

Logger.getLogger(NotePadHM6.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

//Functions - Cope, Cut , Paste

//mECopy.addActionListener (new ActionListener () {

if (menuitem.equals (mECopy)) {

replaced = TextPad.getSelectedText();

}

//mECut.addActionListener (new ActionListener () {

if (menuitem.equals (mECut)) {

//replaced = TextPad.getSelectedText();

// TextPad.removeTextListener((TextListener) ae);

if(TextPad.getSelectedText().length() != 0){

}

replaced = TextPad.getSelectedText();

str = TextPad.getText();

int i = TextPad.getSelectionStart();

int p = TextPad.getSelectionEnd();

TextPad.setText(str.replace(replaced, ""));

}

//mEPaste.addActionListener (new ActionListener () {

if (menuitem.equals(mEPaste)){

Integer start = TextPad.getSelectionStart();

TextPad.insert(replaced, start);

}

}

}

}

**Home Work 6**

package homeworkthreads;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.Choice;

/\*\*

\*

\* @author Axel

\*/

public abstract class HomeWorkThreads extends Frame implements ActionListener, ItemListener {

//Purvonachalno sustoqnie na nishkite

boolean runningthr1 = false;

boolean runningthr2 = false; // pri na tiskane na button star preminavat v true

int result1 = 0; //Shte ni posluji za subirane na rezultite

int result2 = 0;

//Elementi na applet-a

Panel leftPanel = new Panel ();

Panel rightPanel = new Panel ();

TextField tAThread1 = new TextField (20);

TextField tAThread2 = new TextField (20);

//String който ще ни послужи за попълване на List-a / Step

//private static String [] steper = {"1","2","3","5","7","9"};

Choice step1 = new Choice ();

Choice step2 = new Choice ();

Button startT1 = new Button ("Start");

Button stopT1 = new Button ("Stop");

Button startT2 = new Button ("Start");

Button stopT2 = new Button ("Stop");

Label LabelThread1 = new Label ("Thread1");

Label LabelThread2 = new Label ("Thread2");

Label steps1 = new Label ("Step: ");

Label steps2 = new Label ("Step: ");

Label empty = new Label ("");

public HomeWorkThreads () {

//Left Panels elements

leftPanel.setLayout (new GridLayout (7,2));

setBackground(Color.GRAY);

leftPanel.setSize(250,100);

leftPanel.add(empty, BorderLayout.NORTH);

leftPanel.add(LabelThread1);

leftPanel.add(empty);

leftPanel.add(tAThread1);

leftPanel.add(empty);

leftPanel.add(steps1, BorderLayout.WEST);

leftPanel.add(step1, BorderLayout.EAST);

leftPanel.add(empty);

leftPanel.add(startT1, BorderLayout.WEST);

startT1.addActionListener(this);

leftPanel.add(stopT1, BorderLayout.EAST);

stopT1.addActionListener (this);

leftPanel.add(empty, BorderLayout.SOUTH);

//Stop Buttons are UnEnabled

//Enable wenn pressed start button

stopT1.setEnabled(false);

stopT2.setEnabled(false);

//Rights Panels elements

rightPanel.setLayout (new GridLayout (7,2));

setBackground(Color.GRAY);

rightPanel.setSize(250,100);

rightPanel.add(empty, BorderLayout.NORTH);

rightPanel.add(LabelThread2);

rightPanel.add(empty);

rightPanel.add(tAThread2);

rightPanel.add(empty);

rightPanel.add(steps2);

rightPanel.add(step2);

rightPanel.add(empty);

rightPanel.add(startT2, new FlowLayout());

startT2.addActionListener (this);

rightPanel.add(stopT2, new FlowLayout());

stopT2.addActionListener (this);

rightPanel.add(empty, BorderLayout.SOUTH);

//Zapulvane na Spisucite s tqhnata stupka

//step1.addItemListener(this);

step1.add("1");

step1.add("2");

step1.add("3");

step1.add("5");

step1.add("7");

step1.add("9");

//step2.addItemListener(this);

step2.add("1");

step2.add("2");

step2.add("3");

step2.add("5");

step2.add("7");

step2.add("9");

add(leftPanel, BorderLayout.WEST);

add(rightPanel, BorderLayout.EAST);

pack();

setTitle("F65063");

setSize(500,300);

setLocationRelativeTo(null);

setBackground(Color.GRAY);

setResizable(false);

setVisible(true);

//Exit na programata

addWindowListener (new WindowAdapter() {

public void windowClosing(WindowEvent e){

System.exit(0);

}});

final threadRunner lFiller = new threadRunner(this, 1);

lFiller.start();lFiller.suspend();

final threadRunner rFiller = new threadRunner(this, 2);

rFiller.start();rFiller.suspend();

startT1.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent ae) {

stopT1.setEnabled (true);

startT1.setEnabled (false);

runningthr1 = true;

}

});

startT2.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent ae) {

stopT2.setEnabled (true);

startT2.setEnabled (false);

runningthr2 = true;

}

});

stopT1.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent ae) {

startT1.setEnabled (true);

stopT1.setEnabled (false);

runningthr1 = false;

}

});

stopT2.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent ae) {

startT2.setEnabled (true);

stopT2.setEnabled (false);

runningthr2 = false;

}

});

}

/\*public void main(String[] args) {

HomeWorkThreads homework = new HomeWorkThreads() {

public void actionPerformed(ActionEvent ae) {

if (ae.equals(startT1)){

stopT1.setEnabled (true);

startT1.setEnabled (false);

runningthr1 = true;

}

if (ae.equals(startT2)) {

stopT2.setEnabled (true);

startT2.setEnabled (false);

runningthr2 = true;

}

if (ae.equals(stopT1)) {

startT1.setEnabled (true);

stopT1.setEnabled (false);

runningthr1 = false;

}

if (ae.equals(stopT2)) {

startT2.setEnabled (true);

stopT2.setEnabled (false);

runningthr2 = false;

}

}

public void itemStateChanged(ItemEvent e) {

if (e.equals(e)) {

System.exit (0);

}

}

};

threadRunner threads1 = new threadRunner ();

threadRunner threads2 = new threadRunner ();

threads1.run();

threads2.run();

// TODO code application logic here

}\*/

public void start (){

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package homeworkthreads;

import java.awt.Choice;

import java.awt.TextField;

public class threadRunner implements Runnable {

private threadRunner tr;

private TextField field;

private Choice step;

private int i;

//private final Thread t1, t2;

//Integer x = Integer.valueOf(step1);

//int x = Integer.parseInt(step1);

//getSelectedItem(), getSelectedIndex()

public threadRunner(threadRunner tr, int fieldN) {

this.tr = tr;

if(fieldN == 1) {

field = tr.tAThread1;

step = tr.lStep;

} else {

field = tr.tAThread2;

step = tr.rStep;

}

//t1 = new Thread ();//Buttona startT1

//t2 = new Thread ();//Buttona startT2

}

//Startirane na nishkite i Subirane sus suotvetnata stupka

public void run() {

// if (runningthr1) {

// result1 += Integer.parseInt(steps1.toString()); //steps1 e izbora na stupka ot List1

// }

// if (runningthr2) {

// //result2 += Integer.parseInt(steps2); //steps2 e izbora na stupka ot List2

// result2 += Integer.parseInt(steps2.toString());;

// }

// }

// public void stop () {

// runningthr1 = false;

// runningthr2 = false;

// }

while(i < 15000) {

try {

Thread.sleep(150);

} catch (InterruptedException e) {

e.printStackTrace();

}

if(step.getSelectedItem().equals("1")) {

i += 1;

}

if(step.getSelectedItem().equals("2")) {

i += 2;

}

if(step.getSelectedItem().equals("3")) {

i += 3;

}

if(step.getSelectedItem().equals("5")) {

i += 5;

}

if(step.getSelectedItem().equals("7")) {

i += 7;

}

if(step.getSelectedItem().equals("9")) {

i += 9;

}

String result = String.valueOf(i);

field.setText(result);

}

}

void suspend() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

**Home Work 6**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Axel

\*/

public class HomeWorkThreads2 extends Frame {

private final Panel leftPanel = new Panel();

private final Panel midPanel = new Panel();

private final Panel rightPanel = new Panel();

private final Panel footerPanel = new Panel();

private final TextField txtSource = new TextField (5);

private final TextField txtBuffer = new TextField (5);

private final TextField txtDest = new TextField (5);

private final Button btnStart = new Button ("Start");

private final Button btnStop = new Button ("Stop");

private final Label nishka = new Label ("Нишка");

private final Label danni = new Label ("Данни");

private final Label reader = new Label ("Читател");

private final Label writer = new Label ("Писател");

private final Label lBuffer = new Label ("Buffer");

private final Label empty = new Label ("");

public boolean bla;

public HomeWorkThreads2 () {

Label nishka2 = new Label ("Нишка");

Label danni2 = new Label ("Данни");

Label empty1 = new Label ("");

Label empty2 = new Label ("");

//Left Panel - attributes

leftPanel.setLayout (new GridLayout (4,1));

leftPanel.setBackground (Color.BLUE);

leftPanel.add(nishka);

leftPanel.add(writer);

leftPanel.add(txtSource);

leftPanel.add(danni);

//Middle Panel - attributes

midPanel.setLayout (new GridLayout (4,1));

midPanel.add(empty1);

midPanel.add(empty2);

midPanel.add(txtBuffer);

midPanel.add(lBuffer);

//Right Panel - attributes

rightPanel.setLayout (new GridLayout (4,1));

rightPanel.setBackground (Color.YELLOW);

rightPanel.add(nishka2);

rightPanel.add(reader);

rightPanel.add(txtDest);

rightPanel.add(danni2);

//Footer Panel - attributes

footerPanel.setLayout (new FlowLayout());

footerPanel.add(empty);

footerPanel.add(btnStart);

footerPanel.add(btnStop);

footerPanel.add(empty);

//Adding the Panels on Frame

add(leftPanel, BorderLayout.WEST);

add(midPanel, BorderLayout.CENTER);

add(rightPanel, BorderLayout.EAST);

add(footerPanel, BorderLayout.SOUTH);

//Setting UnEnable button stop and wait to be activate from start Button

btnStop.setEnabled(false);

pack();

setTitle("F65063");

setSize(250,200);

setLocationRelativeTo(null);

setResizable(false);

setVisible(true);

txtBuffer.setFocusable(false);

//Exit na programata

addWindowListener (new WindowAdapter() {

@Override

public void windowClosing(WindowEvent e){

System.exit(0);

}});

btnStart.addActionListener (new banani () {

});

btnStop.addActionListener (new banani(){});

}

public static void main(String[] args) {

HomeWorkThreads2 homework = new HomeWorkThreads2();

//ThreadCreate Producer = new ThreadCreate ("Producer", 1000);

//ThreadCreate Consumer = new ThreadCreate ("Consumer", 1500);

// TODO code application logic here

}

public class banani implements ActionListener{

public void actionPerformed (ActionEvent a){

Button b =(Button) a.getSource();

Quantity2 prod = new Quantity2();

Quantity2 cons = new Quantity2();

if(txtBuffer.getText().length() == 0 && txtSource.getText().length() == 0 && txtDest.getText().length() == 0){

prod.buk(0);

cons.buk(0);

}else if(txtBuffer.getText().length() == 0 && txtSource.getText().length() != 0 && txtDest.getText().length() == 0){

prod.buk(Integer.parseInt(txtSource.getText()));

txtBuffer.setText(txtSource.getText());

txtDest.setText(txtSource.getText());

cons.buk(Integer.parseInt(txtSource.getText()));

}

if(b.equals(btnStart) ){

System.out.println("btnstart");

btnStop.setEnabled(true);

bla = true;

prod.start();

cons.start();

System.out.println("end of start");

}

if(b.equals(btnStop)){

System.out.println("Stop clicked");

prod.stoprequestbanani();

cons.stoprequestbanani();

prod.interrupt();

cons.interrupt();

}

}

} }

//Second class Quantity

import java.util.logging.Level;

import java.util.logging.Logger;

public class Quantity2 extends Thread {

public int a ;

//int p = 0;

public boolean stop ;

public int buk (int p){

a = p;

return a;

}

public void stoprequestbanani(){

//currentThread().interrupt();

stop = true;

//return stop;

System.out.println(stop+"Ëxitni mrusna marula");

}

public void run( ) {

int i = 0;

while(!stop){

try {

this.sleep(300);

System.out.println("thread : " + i);

i++;

if(i == a){

this.stoprequestbanani();

}

} catch (InterruptedException ex) {

Logger.getLogger(Quantity2.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

}