

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

**Факультет управления и информатики в технологических системах**

**Кафедра**[**корпоративных информационных систем и программирования**](https://vsuet.ru/obuchenie/faculties/uits/k_kisip)

**Направление подготовки 09.03.02**

***(шифр)***

**Информационные системы и технологии**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(наименование направления подготовки)*

**Отчет**

**по практическому заданию № 1.**

Выполнил студент гр. У-213

Боровкова Анастасия Дмитриевна

(ф.и.о)

**\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(****подпись****)***

Проверил(а):

Демичев С.Е.

**\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(****подпись****)***

Воронеж – 2023

import  java.awt.geom.\*;

import  java.awt.event.\*;

import  java.io.\*;

import  javax.swing.\*;

import  javax.swing.event.\*;

import  java.awt.image.\*;

import  javax.imageio.\*;

import  javax.swing.filechooser.FileFilter;

public class ImageEdit

{

    // Режим рисования

    int  rezhim=0;

    int  xPad;

    int  xf;

    int  yf;

    int  yPad;

    int  thickness;

    boolean pressed=false;

    // текущий цвет

    Color maincolor;

    MyFrame f;

    MyPanel japan;

    JButton colorbutton;

    JColorChooser tcc;

    // поверхность рисования

    BufferedImage imag;

    // если мы загружаем картинку

    boolean loading=false;

    String fileName;

    public ImageEdit()

    {

        f=new MyFrame("Графический редактор");

        f.setSize(350,350);

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

        maincolor=Color.black;

        JMenuBar menuBar = new  JMenuBar();

        f.setJMenuBar(menuBar);

        menuBar.setBounds(0,0,350,30);

        JMenu fileMenu = new  JMenu("Файл");

        menuBar.add(fileMenu);

        Action loadAction = new  AbstractAction("Загрузить")

        {

           public void actionPerformed(ActionEvent event)

           {

              JFileChooser jf= new  JFileChooser();

              int  result = jf.showOpenDialog(null);

               if(result==JFileChooser.APPROVE\_OPTION)

                {

                  try

                  {

                      // при выборе изображения подстраиваем размеры формы

                      // и панели под размеры данного изображения

                       fileName = jf.getSelectedFile().getAbsolutePath();

                       File iF= new  File(fileName);

                       jf.addChoosableFileFilter(new  TextFileFilter(".png"));

                       jf.addChoosableFileFilter(new  TextFileFilter(".jpg"));

                       imag = ImageIO.read(iF);

                       loading=true;

                       f.setSize(imag.getWidth()+40, imag.getWidth()+80);

                       japan.setSize(imag.getWidth(), imag.getWidth());

                        japan.repaint();

                    } catch (FileNotFoundException ex) {

                        JOptionPane.showMessageDialog(f, "Такого файла не существует");

                    }

                    catch (IOException ex) {

                        JOptionPane.showMessageDialog(f, "Исключение ввода-вывода");

                    }

                  catch (Exception ex) {

                    }

                }

              }

            };

        JMenuItem loadMenu = new  JMenuItem(loadAction);

        fileMenu.add(loadMenu);

        Action saveAction = new  AbstractAction("Сохранить")

        {

           public void actionPerformed(ActionEvent event)

           {

               try

               {

                   JFileChooser jf= new  JFileChooser();

                   // Создаем фильтры  файлов

                   TextFileFilter pngFilter = new TextFileFilter(".png");

                   TextFileFilter jpgFilter = new TextFileFilter(".jpg");

                   if(fileName==null)

                   {

                       // Добавляем фильтры

                        jf.addChoosableFileFilter(pngFilter);

                        jf.addChoosableFileFilter(jpgFilter);

                       int  result = jf.showSaveDialog(null);

                       if(result==JFileChooser.APPROVE\_OPTION)

                       {

                           fileName = jf.getSelectedFile().getAbsolutePath();

                       }

                       }

                   // Смотрим какой фильтр выбран

                   if(jf.getFileFilter()==pngFilter)

                   {

                        ImageIO.write(imag, "png", new  File(fileName+".png"));

                   }

                   else

                   {

                       ImageIO.write(imag, "jpeg", new  File(fileName+".jpg"));

                   }

               }

               catch(IOException ex)

               {

                  JOptionPane.showMessageDialog(f, "Ошибка ввода-вывода");

               }

           }

            };

        JMenuItem saveMenu = new  JMenuItem(saveAction);

        fileMenu.add(saveMenu);

        Action saveasAction = new  AbstractAction("Сохранить как...")

        {

           public void actionPerformed(ActionEvent event)

           {

               try

               {

                   JFileChooser jf= new  JFileChooser();

                   // Создаем фильтры для файлов

                   TextFileFilter pngFilter = new  TextFileFilter(".png");

                   TextFileFilter jpgFilter = new  TextFileFilter(".jpg");

                       // Добавляем фильтры

                        jf.addChoosableFileFilter(pngFilter);

                         jf.addChoosableFileFilter(jpgFilter);

                       int  result = jf.showSaveDialog(null);

                       if(result==JFileChooser.APPROVE\_OPTION)

                       {

                           fileName = jf.getSelectedFile().getAbsolutePath();

                       }

                 // Смотрим какой фильтр выбран

                 if(jf.getFileFilter()==pngFilter)

                   {

                        ImageIO.write(imag, "png", new  File(fileName+".png"));

                   }

                else

                   {

                       ImageIO.write(imag, "jpeg", new  File(fileName+".jpg"));

                   }

               }

               catch(IOException ex)

               {

                  JOptionPane.showMessageDialog(f, "Ошибка ввода-вывода");

               }

           }

            };

        JMenuItem saveasMenu = new  JMenuItem(saveasAction);

        fileMenu.add(saveasMenu);

        japan = new  MyPanel();

        japan.setBounds(30,30,260,260);

        japan.setBackground(Color.white);

        japan.setOpaque(true);

        f.add(japan);

        JToolBar toolbar = new  JToolBar("Toolbar", JToolBar.VERTICAL);

          JButton penbutton = new  JButton(new  ImageIcon("pen.png"));

          penbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                rezhim=0;

              }

            });

          toolbar.add(penbutton);

          JButton brushbutton = new  JButton(new  ImageIcon("brush.png"));

          brushbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                rezhim=1;

              }

            });

          toolbar.add(brushbutton);

          JButton lasticbutton = new JButton(new  ImageIcon("lastic.png"));

          lasticbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                rezhim=2;

              }

            });

          toolbar.add(lasticbutton);

          JButton textbutton = new  JButton(new  ImageIcon("text.png"));

          textbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                rezhim=3;

              }

            });

          toolbar.add(textbutton);

          JButton linebutton = new  JButton(new  ImageIcon("line.png"));

          linebutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                rezhim=4;

              }

            });

          toolbar.add(linebutton);

          JButton elipsbutton = new  JButton(new  ImageIcon("elips.png"));

          elipsbutton.addActionListener(new  ActionListener(){

              public void actionPerformed(ActionEvent event)

              {

                rezhim=5;

              }

            });

          toolbar.add(elipsbutton);

          JButton rectbutton = new  JButton(new  ImageIcon("rect.png"));

          rectbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                rezhim=6;

              }

            });

          toolbar.add(rectbutton);

          toolbar.setBounds(0, 0, 30, 300);

          f.add(toolbar);

          // Тулбар для кнопок

          JToolBar colorbar = new  JToolBar("Colorbar", JToolBar.HORIZONTAL);

          colorbar.setBounds(30, 0, 260, 30);

          colorbutton = new  JButton();

          colorbutton.setBackground(maincolor);

          colorbutton.setBounds(15, 5, 20, 20);

          colorbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                ColorDialog coldi = new  ColorDialog(f,"Выбор цвета");

                coldi.setVisible(true);

              }

            });

          colorbar.add(colorbutton);

          JButton redbutton = new  JButton();

          redbutton.setBackground(Color.red);

          redbutton.setBounds(40, 5, 15, 15);

          redbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.red;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(redbutton);

          JButton orangebutton = new  JButton();

          orangebutton.setBackground(Color.orange);

          orangebutton.setBounds(60, 5, 15, 15);

          orangebutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.orange;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(orangebutton);

          JButton yellowbutton = new  JButton();

          yellowbutton.setBackground(Color.yellow);

          yellowbutton.setBounds(80, 5, 15, 15);

          yellowbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.yellow;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(yellowbutton);

          JButton greenbutton = new  JButton();

          greenbutton.setBackground(Color.green);

          greenbutton.setBounds(100, 5, 15, 15);

          greenbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.green;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(greenbutton);

          JButton bluebutton = new JButton();

          bluebutton.setBackground(Color.blue);

          bluebutton.setBounds(120, 5, 15, 15);

          bluebutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.blue;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(bluebutton);

          JButton cyanbutton = new  JButton();

          cyanbutton.setBackground(Color.cyan);

          cyanbutton.setBounds(140, 5, 15, 15);

          cyanbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.cyan;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(cyanbutton);

          JButton magentabutton = new  JButton();

          magentabutton.setBackground(Color.magenta);

          magentabutton.setBounds(160, 5, 15, 15);

          magentabutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.magenta;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(magentabutton);

          JButton whitebutton = new  JButton();

          whitebutton.setBackground(Color.white);

          whitebutton.setBounds(180, 5, 15, 15);

          whitebutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.white;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(whitebutton);

          JButton blackbutton = new  JButton();

          blackbutton.setBackground(Color.black);

          blackbutton.setBounds(200, 5, 15, 15);

          blackbutton.addActionListener(new  ActionListener()

            {

              public void actionPerformed(ActionEvent event)

              {

                maincolor = Color.black;

                  colorbutton.setBackground(maincolor);

              }

            });

          colorbar.add(blackbutton);

          colorbar.setLayout(null);

          f.add(colorbar);

          tcc = new  JColorChooser(maincolor);

          tcc.getSelectionModel().addChangeListener(new  ChangeListener()

          {

          public void stateChanged(ChangeEvent e)

          {

            maincolor = tcc.getColor();

            colorbutton.setBackground(maincolor);

          }

          });

          japan.addMouseMotionListener(new  MouseMotionAdapter()

                  {

                      public void mouseDragged(MouseEvent e)

                      {

                          if (pressed==true)

                          {

                          Graphics g = imag.getGraphics();

                          Graphics2D g2 = (Graphics2D)g;

                          // установка цвета

                          g2.setColor(maincolor);

                          switch (rezhim)

                          {

                              // карандаш

                              case 0:

                                  g2.drawLine(xPad, yPad, e.getX(), e.getY());

                                  break;

                              // кисть

                              case 1:

                                  g2.setStroke(new  BasicStroke(3.0f));

                                  g2.drawLine(xPad, yPad, e.getX(), e.getY());

                                  break;

                               // ластик

                              case 2:

                                   g2.setStroke(new  BasicStroke(3.0f));

                                   g2.setColor(Color.WHITE);

                                   g2.drawLine(xPad, yPad, e.getX(), e.getY());

                                    break;

                          }

                          xPad=e.getX();

                          yPad=e.getY();

                          }

                          japan.repaint();

                      }

                  });

          japan.addMouseListener(new  MouseAdapter()

                  {

                     public void mouseClicked(MouseEvent e) {

                     Graphics g = imag.getGraphics();

                     Graphics2D g2 = (Graphics2D)g;

                     // установка цвета

                          g2.setColor(maincolor);

                          switch (rezhim)

                          {

                              // карандаш

                              case 0:

                                  g2.drawLine(xPad, yPad, xPad+1, yPad+1);

                                  break;

                              // кисть

                              case 1:

                                  g2.setStroke(new  BasicStroke(3.0f));

                                  g2.drawLine(xPad, yPad, xPad+1, yPad+1);

                                  break;

                               // ластик

                              case 2:

                                  g2.setStroke(new  BasicStroke(3.0f));

                                   g2.setColor(Color.WHITE);

                                   g2.drawLine(xPad, yPad, xPad+1, yPad+1);

                              break;

                              // текст

                              case 3:

                                  // устанавливаем фокус для панели,

                                  // чтобы печатать на ней текст

                                  japan.requestFocus();

                              break;

                          }

                          xPad=e.getX();

                          yPad=e.getY();

                          pressed=true;

                          japan.repaint();

                   }

                     public void mousePressed(MouseEvent e) {

                         xPad=e.getX();

                          yPad=e.getY();

                          xf=e.getX();

                          yf=e.getY();

                          pressed=true;

                        }

                    public void mouseReleased(MouseEvent e) {

                        Graphics g = imag.getGraphics();

                        Graphics2D g2 = (Graphics2D)g;

                        // установка цвета

                          g2.setColor(maincolor);

                        // Общие рассчеты для овала и прямоугольника

                        int  x1=xf, x2=xPad, y1=yf, y2=yPad;

                                  if(xf>xPad)

                                  {

                                     x2=xf; x1=xPad;

                                  }

                                  if(yf>yPad)

                                  {

                                     y2=yf; y1=yPad;

                                  }

                        switch(rezhim)

                        {

                             // линия

                              case 4:

                                 g.drawLine(xf, yf, e.getX(), e.getY());

                                  break;

                              // круг

                              case 5:

                                  g.drawOval(x1, y1, (x2-x1), (y2-y1));

                                  break;

                                  // прямоугольник

                              case 6:

                                  g.drawRect(x1, y1, (x2-x1), (y2-y1));

                                  break;

                        }

                        xf=0; yf=0;

                        pressed=false;

                        japan.repaint();

                    }

                  });

        japan.addKeyListener(new  KeyAdapter()

                {

                    public void keyReleased(KeyEvent e)

                    {

                        // устанавливаем фокус для панели,

                       // чтобы печатать на ней текст

                        japan.requestFocus();

                    }

                    public void keyTyped(KeyEvent e)

                    {

                        if(rezhim==3){

                        Graphics g = imag.getGraphics();

                        Graphics2D g2 = (Graphics2D)g;

                        // установка цвета

                          g2.setColor(maincolor);

                        g2.setStroke(new  BasicStroke(2.0f));

                         String str = new  String("");

                         str+=e.getKeyChar();

                        g2.setFont(new  Font("Arial", 0, 15));

                        g2.drawString(str, xPad, yPad);

                        xPad+=10;

                        // устанавливаем фокус для панели,

                        // чтобы печатать на ней текст

                        japan.requestFocus();

                        japan.repaint();

                        }

                    }

                });

        f.addComponentListener(new  ComponentAdapter() {

                public void componentResized(java.awt.event.ComponentEvent evt) {

                    // если делаем загрузку, то изменение размеров формы

                    // отрабатываем в коде загрузки

                   if(loading==false)

                   {

                    japan.setSize(f.getWidth()-40, f.getHeight()-80);

                    BufferedImage tempImage = new  BufferedImage(japan.getWidth(), japan.getHeight(), BufferedImage.TYPE\_INT\_RGB);

                             Graphics2D d2 = (Graphics2D) tempImage.createGraphics();

                        d2.setColor(Color.white);

                        d2.fillRect(0, 0, japan.getWidth(), japan.getHeight());

                    tempImage.setData(imag.getRaster());

                    imag=tempImage;

                    japan.repaint();

                   }

                     loading=false;

                }

                });

        f.setLayout(null);

        f.setVisible(true);

    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new  Runnable() {

      public void run() {

        new  ImageEdit();

      }

    });

    }

    class ColorDialog extends JDialog

    {

      public ColorDialog(JFrame owner, String title)

      {

        super(owner, title, true);

        add(tcc);

        setSize(200, 200);

      }

    }

     class MyFrame extends JFrame

     {

         public void paint(Graphics g)

         {

             super.paint(g);

         }

         public MyFrame(String title)

         {

             super(title);

         }

     }

     class MyPanel extends JPanel

     {

         public MyPanel()

         { }

       public void paintComponent (Graphics g)

          {

            if(imag==null)

             {

                 imag = new  BufferedImage(this.getWidth(), this.getHeight(), BufferedImage.TYPE\_INT\_RGB);

                 Graphics2D d2 = (Graphics2D) imag.createGraphics();

                 d2.setColor(Color.white);

                 d2.fillRect(0, 0, this.getWidth(), this.getHeight());

             }

             super.paintComponent(g);

             g.drawImage(imag, 0, 0,this);

          }

     }

     // Фильтр картинок

     class TextFileFilter extends FileFilter

     {

         private String ext;

         public TextFileFilter(String ext)

         {

             this.ext=ext;

         }

         public boolean accept(java.io.File file)

         {

              if (file.isDirectory()) return true;

              return (file.getName().endsWith(ext));

         }

         public String getDescription()

         {

              return "\*"+ext;

         }

     }

}