**Тест №7. Работа с графикой.**

1. Метод public abstract void setColor (Color c) {…}…

- выводит цвет рисунка;

- устанавливает размеры всех компонент;

- у**станавливает цвет для рисования в контексте отображения;**

- нет верного ответа. Работа с графикой. Определение цвета.

1. Для работы со шрифтом используется:

**- public Font (String name, int style, int size)**

- public Image (String name, int style, int size)

- public Graphic (String name, int style, int size)

- public Polygon (String name, int style, int size) Работа с графикой. Работа со шрифтами.

1. C помощью какого метода выполняется загрузка растрового изображения?

- public Graphic getGraphic (URL url, String name)

**- public Image getImage (URL url, String name)**

- public Graphic setGraphic (URL url, String name)

- public Image setImage (URL url, String name) Работа с графикой. Использование графических файлов.

1. Выберите рисунки, соответствующие результату выполнения кода программы:

import java.awt.\*;

import javax.swing.\*;

public class Window{

public static void main(String[] args) {

JFrame w=new JFrame("Окно с изображением");

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

w.setLayout(new BorderLayout(1,1));

class Canvas extends JComponent{

public void paintComponent(Graphics g){

super.paintComponents(g);

Graphics2D g2d=(Graphics2D)g;

g2d.setPaint(Color.GREEN);

g2d.drawRect(100, 100, 80, 20);

g2d.setPaint(Color.RED);

g2d.drawString("Привет мир", 150, 150);

g2d.setColor(Color.blue);

g2d.fillOval(200, 50, 50, 20);

super.repaint();

}

}

Canvas canv=new Canvas();

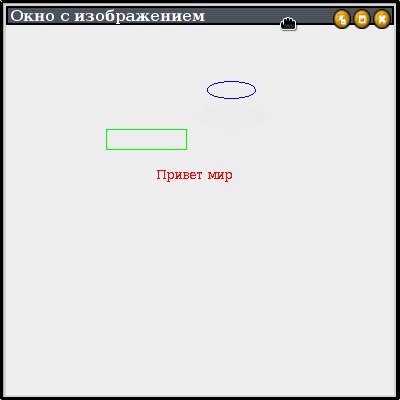
w.add(canv);

w.setVisible(true);

}}

1. 2. 3.



- **1;**

- 2;

- 3;

- нет верного ответа.

1. Выберите верные конструкции создания объектов, устанавливающие цвет:

- Color pureRed = new Color(255, 0, 0);

- Color someColor = new Color(0.5f, 0.4f, 0.95f);

- Color c1 = new Color(0xFF8F48);

- Color pureBluePr = new Color(0, 0, 255,125);

**- все варианты.** Работа с графикой. Выполнение программы.

1. Выберите функцию, график которой удовлетворяет коду программы:

import java.awt.\*;

import java.awt.geom.Line2D;

import javax.swing.\*;

import java.lang.Math;

public class Main

{public static void main(String[] args)

{DrawFrame frame = new DrawFrame();

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

frame.setVisible(true);}

}

class DrawFrame extends JFrame

{public DrawFrame()

public static final int DEFAULT\_WIDTH = 400;

public static final int DEFAULT\_HEIGHT = 400;

}

class DrawPanel extends JPanel

{public void paintComponent(Graphics g)

{super.paintComponent(g);

Graphics2D g2 = (Graphics2D)g;

double shX = 150;

double shY = 100;

for(int x = -6; x < 6; ++x)

g2.draw

(new Line2D.Double (x + shX, DrawFrame.DEFAULT\_HEIGHT - Math.pow(3, -x - 1) + shY),x + 1 + shX, DrawFrame.DEFAULT\_HEIGHT - (Math.pow(3, -x - 2) + shY)));

}

}

**- y=(2/3)^(x+2);**

- y=3^(-х-1)+sin(x+1);

- y=3^(-х-2)+sin(x+1);

- нет верного ответа;

- все варианты. Работа с графикой. Построение функций.

1. Метод public void setPaintMode() устанавливает:

- текущий цвет рисования;

- **режимы рисования**;

- текущий шрифт;

- панель для рисования. Работа с графикой. Возможности Java 2D.

1. Выберите методы рисования фигур:

- public void fillRect (int x, int y, int points)

- public abstract void setColor (Color c)

- **public abstract void drawLine (int x1, int y1, int х2, int y2)**

**- public abstract void drawPolyline (int[] x, int[] y, int points)**

- public abstract void fillLine (int x1, int y1, int х2, int y2) Работа с графикой. Методы класса Graphics для рисования фигур.

1. Выберите рисунок, определяющий результат выполнения кода программы:

import java.awt.\*;

import j ava.awt.event.\*;

class GraphTest extends Frame{

GraphTest(String s) {

super(s);

setBounds(0, 0, 500, 300);

setVisible(true); }

public void paint(Graphics g){

Dimension d = getSizeO;

int dx = d.width/20, dy-d.height/20;

g.drawRect(dx, dy+20, d.width-2\*dx, d.height-2\*dy-20);

g.drawRoundRect(2\*dx,2\*dy+20,d.width-4\*dx,d.height-4\*dy-20,dx, dy);

g.fillArctd.width/2-dx, d.height-2\*dy+1, 2\*dx, dy-1, 0, 360);

g.drawArctd.width/2-3\*dx, d.height-3\*dy/2-5, dx, dy/2, 0, 360);

g.drawArctd.width/2+2\*dx, d.height-3\*dy/2-5, dx, dy/2, 0, 360);

Font fl = new Font("Serif", Font.BOLD(Font.ITALIC, 2\*dy);

Font f2 = new Font ("Serif", Font.BOLD, 5\*dy/2);

FontMetrics fml = getFontMetrics(fl);

FontMetrics fm2 = getFontMetrics(f2);

String s1 = "Всякая последняя ошибка";

String s2 = "является предпоследней.";

String s3 = "Закон отладки";

int firstLine = d.height/3;

int nextLine = fml.getHeight();

int secondLine = firstLine + nextLine/2;

g.setFont(f2);

g.drawstring(s3, (d.width-fm2.stringWidth(s3))/2, firstLine);

g.drawLine(d.width/4, secondLine-2, 3\*d.width/4, secondLine-2);

g.drawLine(d.width/4, secondLine—1, 3\*d.width/4, secondLine-1);

g.drawLine(d.width/4, secondLine, 3\*d.width/4, secondLine);

g.setFont(fl);

g.drawstring(s1,(d.width-fml.stringWidth(si))12, firstLine+2\*nextLine);

g.drawString(s2,(d.width-fml.stringWidth(s2))/2,firstLine+3\*nextLine); }

public static void main(String[] args){

GraphTest f = new GraphTest(" Пример рисования");

f.addWindowListener(new WindowAdapter(){

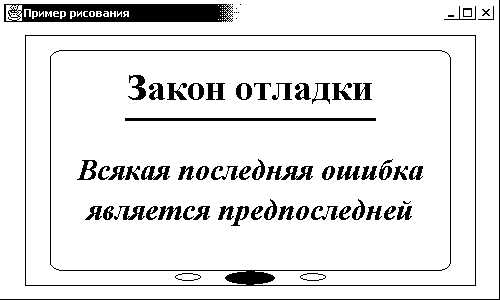
public void windowClosing(WindowEvent ev){

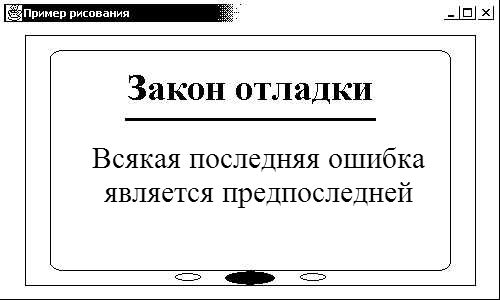
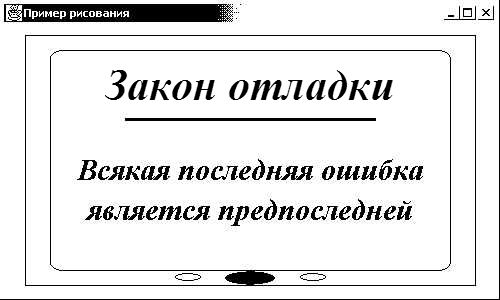
System.exit(0); }});

}

}

1. 2. 3.





**- 1;**

- 2;

- 3;

- нет верного ответа. Работа с графикой. Свойства шрифтов

1. Метод public abstract void clipRect(int x, int y, int width, int height)…

- предназначен для рисования прямоугольника;

**- задает область ограничения для рисования;**

- предназначен для рисования заполненного овала;

- определяет систему координат. Работа с графикой. Базовые методы класса Graphics.