

一. 简答题

- 1
- 2
- 3
- 6

Java语言实习5 GUI与Applet应用2023

题量: 6 满分: 100

作答时间: 2023-10-09 10:18 至 2023-10-21 00:18

一. 简答题 (共6题)

1. (简答题)

1、修改类名，使用不同字体绘制字符串。以下程序设置了3种字体，试自行编写HTML文件，注意所有字符要显示在屏幕内；分析以下代码和HTML文件的运行过程及结果。

```
import java.applet.Applet;
import java.awt.*;
public class Exam4_4 extends Applet
{
    public void paint(Graphics g)
    {
        Font fnt1,fnt2,fnt3;
        fnt1=new Font("Serif",Font.BOLD,20);
        fnt2=new Font("Monospaced",Font.BOLD+Font.ITALIC,24);
        fnt3=new Font("SansSerif",Font.PLAIN,16);
        g.setFont(fnt1);
        g.drawString("Serif 20 point BOLD",30,20);
        g.setFont(fnt3);
        g.drawString("SansSerif 16 point PLAIN",30,50);
        g.setFont(fnt2);
        g.drawString("Monospaced 24 point BOLD+ITALIC",30,80);
        String str,name;
        int size,style;
        size=fnt2.getSize();
        style=fnt1.getStyle();
        name=fnt2.getName();
        str=name+" "+style+" "+size;
        g.drawString(str,30,110);
    }
}
```

类名姓名首字母_String。

得分标准:

- (1) 可不在html实现，GUI和Applet均可，但需编写html，代码截图即可1分，需另有文字说明html文件中参数意义1分，无文字本小题0分；
- (2) 代码截图1分，结果截图1分；
- (3) 代码运行分析6分，需有功能的文字分析，如无或过简0分。

我的答案:

代码行截图:

```
{
    public void paint(Graphics g)
    {
        Font fnt1,fnt2,fnt3;
        fnt1=new Font("Serif",Font.BOLD,20);
        fnt2=new Font("Monospaced",Font.BOLD+Font.ITALIC,24);
        fnt3=new Font("SansSerif",Font.PLAIN,16);
        g.setFont(fnt1);
        g.drawString("SansSerif 16 point PLAIN",30,20);
        g.setFont(fnt3);
        g.drawString("SansSerif 16 point PLAIN",30,50);
        g.setFont(fnt2);
        g.drawString("Monospaced 24 point BOLD+ITALIC",30,80);
        String str,name;
        int size,style;
        size=fnt2.getSize();
        style=fnt1.getStyle();
        name=fnt2.getName();
        str=name+" "+style+" "+size;
        g.drawString(str,30,110);
    }
}
```

一. 简答题

1

2

3

6

代码文字分析说明:

1.代码定义了三个字体对象fnt1, fnt2 和 fnt3, 每一个都使用不同的字体名称、样式和大小进行初始化。这些字体对象分别代表"Serif"、"Monospaced" 和 "SansSerif" 字体。

2.g.setFont(fnt1)将绘图上下文的字体设置为 fnt1, 并使用 g.drawString绘制字符串 "Serif 20 point BOLD" 在坐标 (30, 50) 处。

3.g.setFont(fnt3)将字体设置为 fnt3, 并使用 g.drawString绘制字符串 "SansSerif 16 point PLAIN" 在坐标 (30, 80) 处。

4.g.setFont(fnt2)将字体设置为 fnt2, 并使用 g.drawString绘制字符串 "Monospaced 24 point BOLD+ITALIC" 在坐标 (30, 110) 处。

5.代码获取了fnt2的字体信息, 包括字体名称、样式和大小, 将这些信息存储在 name, style和 size变量中, 然后将它们连接成一个字符串并存储在str变量中。

6.使用g.drawString将包含字体信息的字符串 "Monospaced 24 point BOLD+ITALIC" 绘制在坐标 (30, 140) 处。

html文件:

```
<!DOCTYPE html>
<html>
<head>
  <title>Java小程序示例</title>
</head>
<body>
  <applet code="cxy_String.class" width="400" height="200">
    您的浏览器不支持Java小程序。
  </applet>
</body>
</html>
```

html文件参数分析:

applet: 这是HTML标签, 用于嵌入Java小程序 (applet) 。

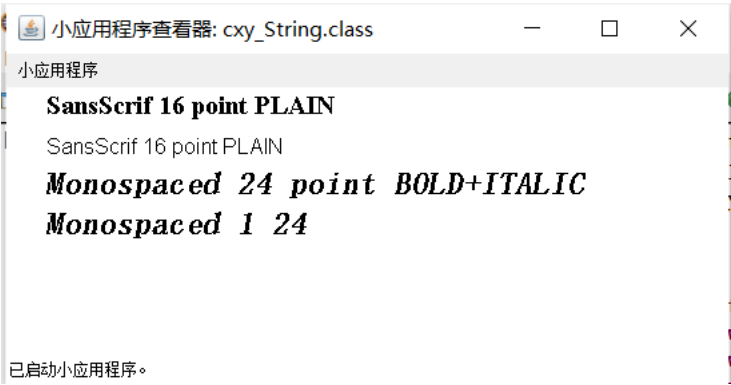
code="cxy_String.class": 这是<applet>标签的一个属性, 指定要运行的Java小程序的类文件。在这里, 它引用了名为cxy_String.class的Java类文件, 该文件包含您的小程序。

width="400": 这是<applet>标签的一个属性, 设置小程序的宽度为400像素。

作业详情

applet内的文本：“您的浏览器不支持Java小程序。”：这是在不支持Java小程序的浏览器上显示的替代文本。如果浏览器支持Java小程序，它将执行小程序的内容；否则将显示这段文本。

运行结果截图：



一. 简答题

- 1
- 2
- 3
- 6

2. (简答题)

2、设置不同颜色显示字符串。将“学号+姓名大写拼音”的字符串分别用不同字体的粗体、斜体、常规格式展示在不同行，字号分别为 30，40，50，首字符缩进 30 字符；并为这三行字符串赋予不同颜色。输出的字符串包括学号+姓名+获取的字体（名称+类型）+获取的字号+获取的颜色，空格隔开。如：
2018102130022 ZHANGSHAN Serif PLAIN 30 r=100, g=255, b=255

类名姓名首字母_DrawString，将 ZHANGSHAN 改为姓名全称小写拼音，其他按要求进行。

- (1) 代码与结果截图3分，代码必须按要求进行，如错误或直接赋值，此大题0分。
(2字符串15分，每行5分，字体、格式、字号、三个灰度值必须采用方法或属性来获取相应值，每处不符合要求扣1分，扣完为止。

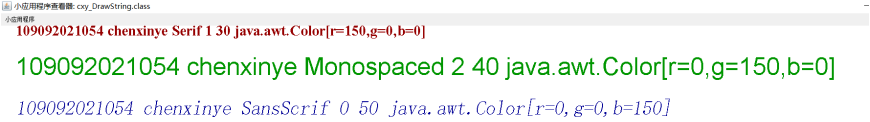
我的答案：

代码行：

```
import java.applet.Applet;
import java.awt.*;
public class cxy_DrawString extends Applet {
    public void paint(Graphics g)
    {
        // 创建三个自定义颜色对象
        Color customcolor1=new Color(150,0,0);
        Color customcolor2=new Color(0,150,0);
        Color customcolor3=new Color(0,0,150);
        //创建三个不同的字体对象
        Font fnt1,fnt2,fnt3;
        fnt1=new Font("Serif",Font.BOLD,30);
        fnt2=new Font("Monospaced",Font.ITALIC,40);
        fnt3=new Font("SansSerif",Font.PLAIN,50);

        // 绘制第一行字符串，包括学号、姓名、字体信息、字号和颜色信息
        g.setFont(fnt1);
        g.setColor(customcolor1);
        g.drawString("109092021054 chenxinye"+ " "+fnt1.getName()+" "+fnt1.getStyle()+" "+fnt1.getSize()+" "+customcolor1,30,20);
        // 绘制第二行字符串，包括学号、姓名、字体信息、字号和颜色信息
        g.setFont(fnt2);
        g.setColor(customcolor2);
        g.drawString("109092021054 chenxinye"+ " "+fnt2.getName()+" "+fnt2.getStyle()+" "+fnt2.getSize()+" "+customcolor2,30,100);
        // 绘制第三行字符串，包括学号、姓名、字体信息、字号和颜色信息
        g.setFont(fnt3);
        g.setColor(customcolor3);
        g.drawString("109092021054 chenxinye"+ " "+fnt3.getName()+" "+fnt3.getStyle()+" "+fnt3.getSize()+" "+customcolor3,30,180);
    }
}
```

代码结果：



3. (简答题)

用drawLine()方法绘制一个网格，类名为姓名首字母_完整学号，要求如下：

- (1) 文本框或控制台输入学号，设置网格行个数为学号前2位，列个数为末2位，当行大于列，行列数互换。
(2) 每个网格高40字符，宽20字符，for循环根据行列号绘制网格线。
(3) 网线颜色rgb分别设置为：学号1-3位，4-6位，10-12位，奇数赋值0，偶数赋值128。如rgb均为128，任选一项设置为0。

得分标准：

- (1) 类名、代码和结果截图1分，如类名错误、无截图、截图不完整、代码和截图结果不符本大题0分。

作业详情

- (4) 行间距1分，列间距1分。
- (5) r, g, b各1分，调整1分。
- (6) 使用for循环且结果正确2分，不使用for循环本大题0分。

提示：

本题考核字符串截取，数据格式转换，Math方法，画线方法。
以红色为例：用substring截取输入学号的1-3位，用Integer.parseInt转为int型，整除用%，三目判断赋值。
画线方法drawLine(x0,y0,x1,y1)，起点坐标x0,y0，终点坐标x1,y1；
画行列线时注意：一行有row个网格即要画row+1条线，一列有line个网格即要画line+1条线。

一. 简答题

1

2

3

6

我的答案：

代码行：

```
import java.applet.Applet;
import javax.swing.JOptionPane;
import java.awt.*;
import java.util.Scanner;

public class cxy_109092021054 extends Applet {
    public void paint(Graphics g) {
        // 使用对话框来输入学号
        String studentNumber = JOptionPane.showInputDialog("请输入您的学号：");
        if (studentNumber == null) {
            return;
        }

        // 从学号中提取前两位和后两位
        int head = Integer.parseInt(studentNumber.substring(0, 2));
        int last = Integer.parseInt(studentNumber.substring(studentNumber.length() - 2));

        // 确保 head 存储的是较小的值，last 存储的是较大的值
        if (head > last) {
            int temp = head;
            head = last;
            last = temp;
        }

        // 定义行数、列数、以及网格的宽度和高度
        int rows = last;
        int columns = head;
        int gridWidth = 20;
        int gridHeight = 40;

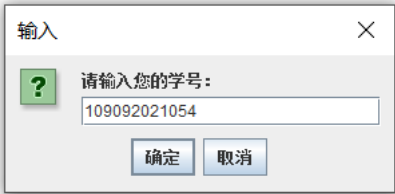
        // 从学号中提取特定位置的数字
        int firstDigits = Integer.parseInt(studentNumber.substring(0, 3));
        int middleDigits = Integer.parseInt(studentNumber.substring(3, 6));
        int lastDigits = Integer.parseInt(studentNumber.substring(9, 12));

        // 根据学号的属性来确定颜色
        int red, green, blue;
        if (firstDigits == 128 && middleDigits == 128 && lastDigits == 128) {
            red = 0; // 如果rgb均为128，将红色分量设置为0
            green = 128;
            blue = 128;
        }
        else{
            red = (firstDigits % 2 == 0) ? 128 : 0;
            green = (middleDigits % 2 == 0) ? 128 : 0;
            blue = (lastDigits % 2 == 0) ? 128 : 0;
        }

        // 设置绘图颜色
        g.setColor(new Color(red, green, blue));

        // 绘制水平网格线
        for (int row = 0; row <= rows; row++) {
            int y = row * gridHeight;
            g.drawLine(0, y, columns * gridWidth, y);
        }
        // 绘制垂直网格线
        for (int col = 0; col <= columns; col++) {
            int x = col * gridWidth;
            g.drawLine(x, 0, x, rows * gridHeight);
        }
    }
}
```

代码结果：



作业详情

[illegible]

因为学号末尾为054，有54列，小程序窗口无法全部展示，换为015展示

一. 简答题

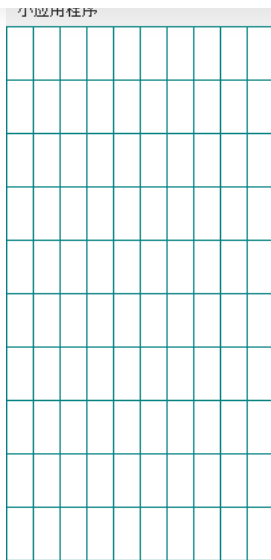
1

2

(11)

6

作业详情



一. 简答题

1

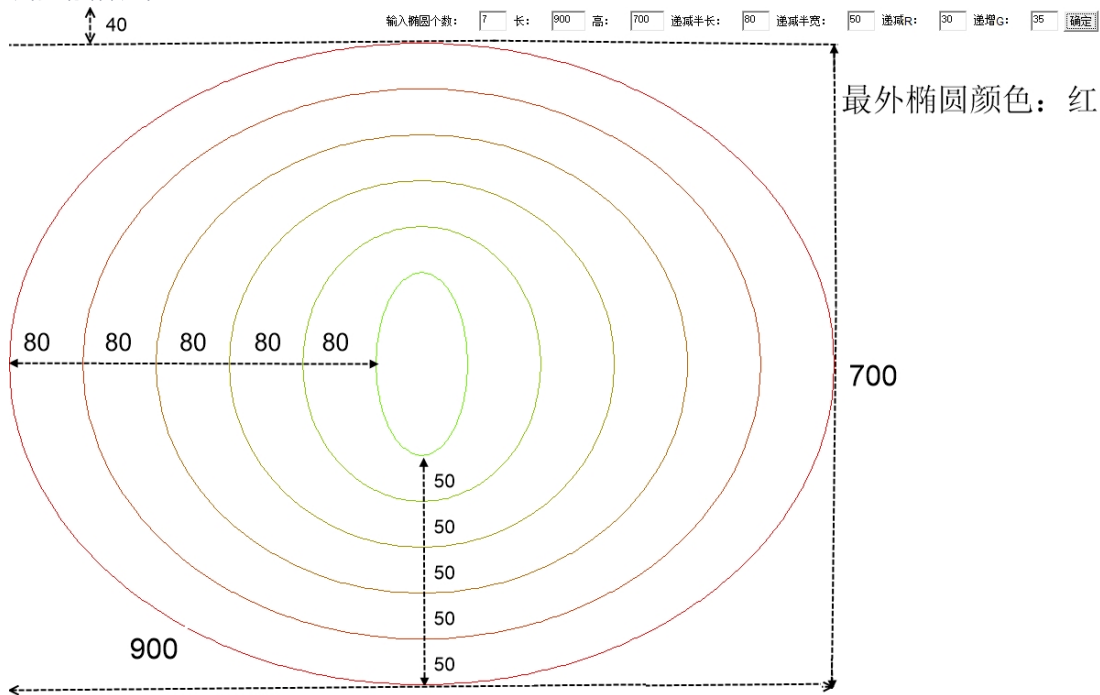
2

3

6

4. (简答题)

现有运行结果如下：



图中虚线和右侧文字仅为提示信息，无需实现。

由用户设置椭圆组个数, 最大椭圆长宽, 递减半长宽, 内侧椭圆颜色r、g变化, 最外椭圆为红色, b保持不变为0; 设置完毕后点击确定, 即可绘制参数设定下的椭圆组。for循环实现。

类名姓名首字符_oval。

得分标准:

- (1) 代码和结果截图2分，如类名有误、代码不清晰、代码不全、结果有误、未按要求，所有小题0分。
- (2) 控件4分，设置2分，事件响应2分，错一处扣1分，扣完为止；
- (3) 椭圆组8分，椭圆个数1分，绘制方法1分，坐标2分，边长2分，颜色2分，错一处扣1分，扣完为止；
- (4) 代码分析4分，必须有注释外的单独文字分析代码运行过程与功能。

提示:

本题涉及到按钮事件，需引用java.awt.event.*;

主类要继承类Applet，继承接口ActionListener: public class ** extends Applet implements ActionListener

参数用文本框TextField, 构造形参为文本框长度, 提示用Label, 构造形参为显示字符串, 按钮用Button, 构造形参为显示字符串:

getText可返回文本框的字符串。
repaint()绘制刷新画面。
init()初始化载入控件，注意要为按钮添加监听器，addActionListener(this)
drawOval空心椭圆，fillOval实心椭圆。参数为（左上角x，左上角y，椭圆长，椭圆高）

一. 简答题

1

2

3

6

我的答案：

代码行截图：

```
1import java.applet.Applet;
2import java.awt.*;
3import java.awt.event.ActionEvent;
4import java.awt.event.ActionListener;
5public class EllipseApplet extends Applet implements ActionListener {
6    // 文本框中输入参数
7    private TextField numEllipsesField, maxEllipseWidthField, maxEllipseHeightField, decrementWidthField, decrementHeightField, innerEllipseRedField, innerEllipseGreenField;
8    private Label numEllipsesLabel, maxEllipseWidthLabel, maxEllipseHeightLabel, decrementWidthLabel, decrementHeightLabel, innerEllipseRedLabel, innerEllipseGreenLabel;
9    private Button drawButton;
10    private int numEllipses, maxEllipseWidth, maxEllipseHeight, colorRed, colorGreen, dishigh, diswide;
11
12    public void init() {
13        // 创建文本框、标签和按钮
14        numEllipsesLabel = new Label("椭圆数量:");
15        numEllipsesField = new TextField(5);
16        maxEllipseWidthLabel = new Label("最大椭圆宽度:");
17        maxEllipseWidthField = new TextField(5);
18        maxEllipseHeightLabel = new Label("最大椭圆高度:");
19        maxEllipseHeightField = new TextField(5);
20        decrementWidthLabel = new Label("递减宽度:");
21        decrementWidthField = new TextField(5);
22        decrementHeightLabel = new Label("递减高度:");
23        decrementHeightField = new TextField(5);
24        innerEllipseRedLabel = new Label("递减内侧椭圆red的变化值:");
25        innerEllipseRedField = new TextField(5);
26        innerEllipseGreenLabel = new Label("递减内侧椭圆green的变化值:");
27        innerEllipseGreenField = new TextField(5);
28
29        drawButton = new Button("绘制");
30
31        // 添加按钮点击事件监听
32        drawButton.addActionListener(this);
33
34        // 将控件添加到Applet
35        add(numEllipsesLabel);
36        add(numEllipsesField);
37        add(maxEllipseWidthLabel);
38        add(maxEllipseWidthField);
39        add(maxEllipseHeightLabel);
40        add(maxEllipseHeightField);
41        add(decrementWidthLabel);
42        add(decrementWidthField);
43        add(decrementHeightLabel);
44        add(decrementHeightField);
45        add(innerEllipseRedLabel);
46        add(innerEllipseRedField);
47        add(innerEllipseGreenLabel);
48        add(innerEllipseGreenField);
49        add(drawButton);
50    }
51
52    public void actionPerformed(ActionEvent e) {
53        if (e.getSource() == drawButton) {
54            // 从文本框中获取参数
55            numEllipses = Integer.parseInt(numEllipsesField.getText());
56            maxEllipseWidth = Integer.parseInt(maxEllipseWidthField.getText());
57            maxEllipseHeight = Integer.parseInt(maxEllipseHeightField.getText());
58            colorRed = Integer.parseInt(innerEllipseRedField.getText());
59            colorGreen = Integer.parseInt(innerEllipseGreenField.getText());
60            dishigh = Integer.parseInt(decrementHeightField.getText());
61            diswide = Integer.parseInt(decrementWidthField.getText());
62            repaint(); // 绘制椭圆
63        }
64    }
65
66    public void paint(Graphics g) {
67        int centerX = getWidth() / 2;
68        int centerY = getHeight() / 2;
69
70        int red = 255; // 红色分量为255
71        int green = 0;
72        int blue = 0;
73
74        for (int i = 0; i < numEllipses; i++) {
75            g.setColor(new Color(red, green, blue));
76            g.drawOval(centerX - (maxEllipseWidth / 2), centerY - (maxEllipseHeight / 2), maxEllipseWidth, maxEllipseHeight);
77
78            // 逐渐减小椭圆的长宽
79            maxEllipseWidth -= dishigh;
80            maxEllipseHeight -= diswide;
81
82            // 调整颜色
83            green += colorRed;
84            red -= colorGreen;
85        }
86    }
87}
```

代码过程分析：

- 1.首先进行用户界面初始化：创建多个标签（Label）和文本框（TextField），以及一个绘制按钮（Button）。用户可以在文本框中输入椭圆的数量、最大椭圆的宽度和高度、内侧椭圆颜色的变化值（红色和绿色），以及椭圆尺寸的递减量。
- 2.设置ActionListener 和 按钮点击事件类实现了ActionListener接口，用于监听按钮点击事件。当用户点击“绘制”按钮后，actionPerformed方法会被调用。
- 3.处理按钮点击事件：在actionPerformed方法中，检查事件源是否是“绘制”按钮。如果是，就开始处理用户的输入。从文本框中获取用户输入的值，包括椭圆数量、椭圆的宽度和高度、内侧椭圆颜色的变化值（红色和绿色），以及椭圆尺寸的递减量。调用repaint()来触发paint方法。
- 4.绘制椭圆：在paint方法中，首先定义了绘制的起始坐标（centerX 和 centerY）以确保椭圆在 Applet 窗口的中心。初始化颜色变量red 为255，green为0，blue为0。通过for循环来绘制用户指定数量的椭圆。绘制椭圆时，使用drawOval方法，并设置当前颜色为Color(red, green, blue)。椭圆的宽度和高度会根据用户设置的递减值逐渐减小。内侧椭圆的颜色将根据用户设置的红色和绿色变化值逐渐变化。

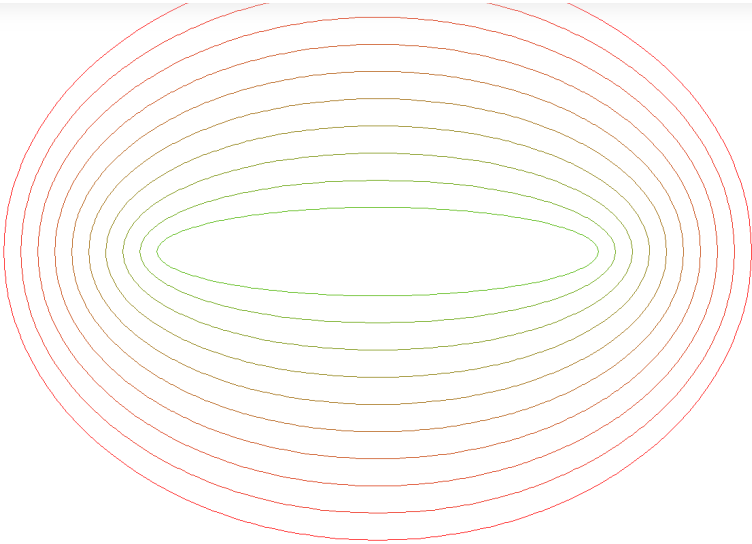
实验结果：

椭圆数量: 10 最大椭圆宽度: 1100 最大椭圆高度: 850 递减宽度: 80 递减高度: 80 递减内侧椭圆red的变化值: 20 递减内侧椭圆green的变化值: 20 绘制

作业详情

一. 简答题

- 1
- 2
- 3
- 6



5. (简答题)

5、绘制一个直方图，要求如下：

- (1) 有 X 轴和 Y 轴，轴有实心箭头，有提示字符；
- (2) Y 轴有 5 条短刻度线，间距 30 字符；
- (3) 直方图有 3 个实心矩形，颜色分别为深蓝、浅蓝和白色；宽均为 30 字符，间距 10 字符，高度分别为 130、80、20 字符。

要求如下：

- (1) 类名姓名首字母_histogram，截图2分，有误所有小题0分；
- (2) X轴的位置2分、箭头2分，Y轴的位置2分、箭头2分，Y轴刻度线4分；提示字符2分；刻度线必须用for循环。
- (3) 矩形形状2分，颜色2分，位置2分。

各要素有误扣2分，扣完为止。

提示：实心箭头fillPolygon，字符drawString，矩形fillRect，轴drawLine，刻度for循环 drawLine，背景色setBackground。

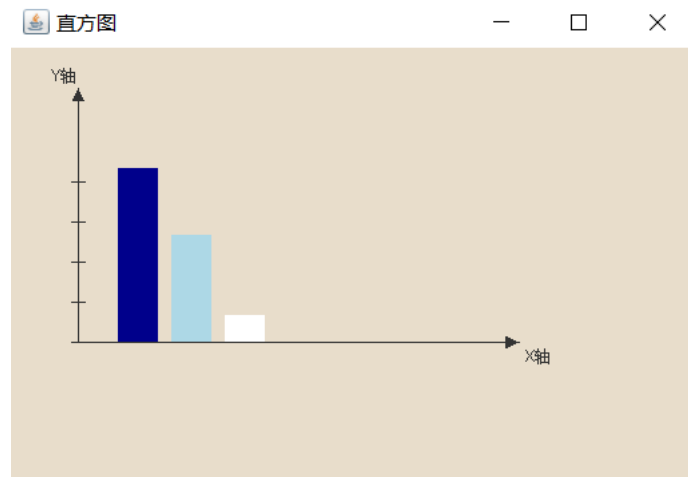
我的答案：

代码行：

作业详情

```
3 import java.awt.Color;
4
5 public class cxy_histogram extends JPanel {
6     public void paintComponent(Graphics g) {
7         super.paintComponent(g);
8
9         // 设置背景和矩形颜色
10        this.setBackground(new Color(232,221,203));
11
12        // 设置Y轴和X轴
13        g.drawLine(50, 220, 50, 30);
14        g.drawLine(50, 220, 380, 220);
15
16        // 绘制Y轴实心箭头
17        int[] yArrowXPoints = {50, 45, 55};
18        int[] yArrowYPoints = {30, 40, 40};
19        g.fillPolygon(yArrowXPoints, yArrowYPoints, 3);
20
21        // 绘制X轴实心箭头
22        int[] xArrowXPoints = {380, 370, 370};
23        int[] xArrowYPoints = {220, 215, 225};
24        g.fillPolygon(xArrowXPoints, xArrowYPoints, 3);
25
26        // 添加提示字符
27        g.drawString("Y轴", 30, 25);
28        g.drawString("X轴", 385, 235);
29
30        // 绘制Y轴刻度线, 间隔30个字符
31        for (int i = 0; i < 5; i++) {
32            int y = 220 - i * 30;
33            g.drawLine(45, y, 55, y);
34        }
35
36        // 绘制直方图的三个矩形, 间隔30个字符
37        g.setColor(new Color(0,0,139));
38        g.fillRect(80, 90, 30, 130);
39        g.setColor(new Color(173,216,230));
40        g.fillRect(120, 140, 30, 80);
41        g.setColor(Color.WHITE);
42
43        public static void main(String[] args) {
44            JFrame frame = new JFrame("直方图");
45            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46            frame.add(new cxy_histogram());
47            frame.setSize(400, 250);
48            frame.setVisible(true);
49        }
50    }
51 }
```

实验结果:



一. 简答题

1

2

3

6

得分标准:

- (1) 本题可不在html中载入, 但需有html的代码截图, applet的代码和结果截图, 需附上Applet的java文件、class文件、音乐素材、图像素材, 打包以附件提交。截图或素材缺失所有小题0分。
- (2) 素材 4分: HTML文件1分, 图片2分, 音乐1分;
- (3) 引用4分: 引用类无误2分, 方法无误2分;
- (4) 素材使用4分: 图像无误2分, 音乐无误2分
- (5) 代码分析4分, 必须有独立分析文字。

各要素有误扣2分, 扣完为止。

一. 简答题

1

2

3

6

我的答案:

代码行:

```
1 import java.applet.Applet;
2 import java.awt.*;
3 import java.applet.AudioClip;
4
5
6 public class last extends Applet {
7     Image image;
8     AudioClip audio;
9
10    public void init() {
11        // 加载图片
12        image = getImage(getCodeBase(), "yasuo.jpg");
13
14        // 加载音频
15        audio = getAudioClip(getCodeBase(), "song.wav");
16    }
17
18    public void paint(Graphics g) {
19        // 在Applet中显示图片
20        g.drawImage(image, 0, 0, this);
21    }
22
23    public void start() {
24        // 循环播放背景音乐
25        audio.loop();
26    }
27
28    public void stop() {
29        // 停止音乐
30        audio.stop();
31    }
32 }
```

代码分析:

当Applet加载时, init方法被调用, 加载图像和音频文件,使用 getImage和 getAudioClip方法从 getCodeBase()获取图像和音频文件的URL。随后, paint方法用于在Applet中显示图像。同时, start方法使用 audio.loop()方法循环播放背景音乐。如果关闭Applet, stop方法会停止音乐的播放。

html代码行:

作业详情

```
<html>
<head>
  <title>Applet示例</title>
</head>
<body>
  <h1>背景音乐和图片</h1>

  <applet code="BackgroundMusicApplet.class" width="400" height
    <param name="image" value="yasuo.jpg">
    <param name="music" value="song.wav">
  </applet>
</body>
</html>
```

代码分析:

运行结果:



109092021054.zip

一. 简答题

- 1
- 2
- 3
- 6