



北京師範大學
BEIJING NORMAL UNIVERSITY

《Java 程序设计》上机实 验报告

一、实验要求

1. 上机之前应做好充分准备，认真思考所需的上机题目，提高上机效率。
2. 独立上机输入和调试自己所编的程序，切忌抄袭、拷贝他人程序。
3. 上机结束后，整理出实验报告。书写报告时，重点放在实验的方法、思路以及总结反思上，以达到巩固课堂学习、提高动手能力的目的。
4. 红字标出部分是本节实验中必须提交截图或者说明的部分，请同学们认真解答。

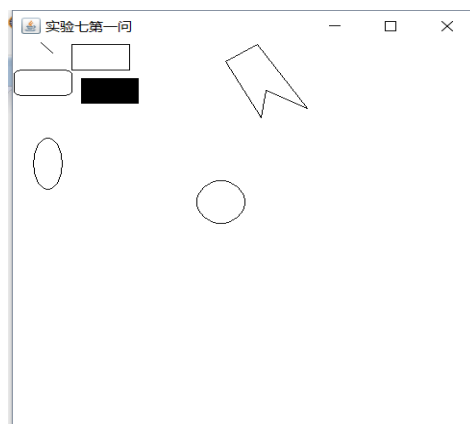
二、实验过程

(对于每个问题，可以包括：

1. 实验步骤，如对某个问题的分析、算法思路；
2. 实验过程，如关键代码。
3. 运行结果，如相关问题运行结果的截图，结果说明等等。)

1. 编写在 Frame 中绘制直线、各种矩形、多边形、圆和椭圆等图形。

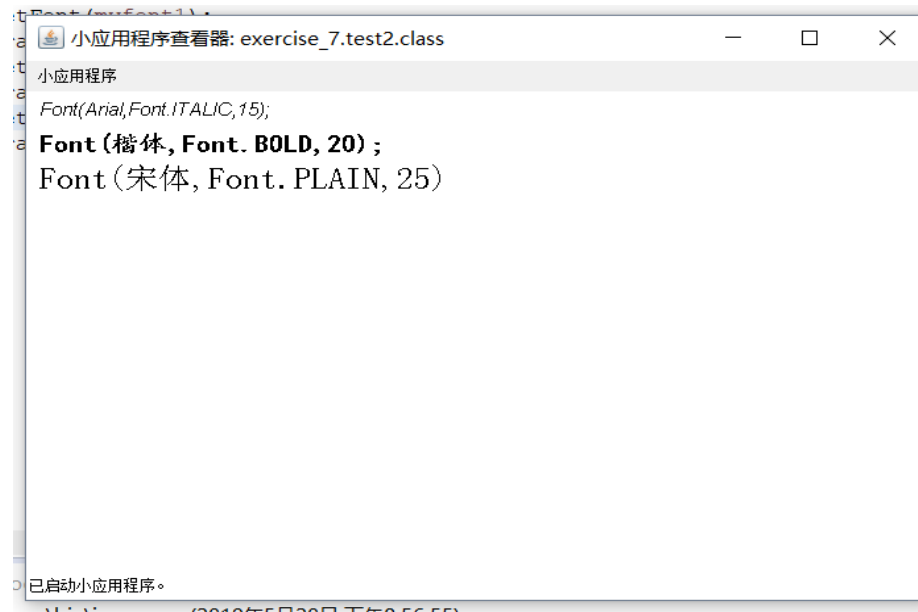
(运行截图+代码截图)



```
1 package exercise_7;
2 import java.awt.*;
3
4
5 public class test1 extends JFrame{
6
7     public void paint(Graphics g){
8         g=this.getGraphics();
9         g.drawLine(10, 10, 50, 50);
10        g.drawRect(70, 40, 60, 30);
11        g.drawRoundRect(10, 70, 60, 30, 15, 10);
12        //g.draw3DRect(80, 80, 60, 30, true);
13        g.fill3DRect(80, 80, 60, 30, true);
14        int Poly1_x[]={230,263,315,272,267};
15        int Poly1_y[]={60,40,115,94,126};
16        g.drawPolygon(Poly1_x, Poly1_y, 5);
17        g.drawOval(30, 150, 30, 60);
18        g.drawOval(200, 200, 50, 50);
19
20    }
21    public static void main(String[] args) {
22        test1 my_frame=new test1();
23        my_frame.setDefaultCloseOperation(3);
24        my_frame.setTitle("实验七第一问");
25        my_frame.setSize(500,500);
26        my_frame.setVisible(true);
27
28    }
29 }
30 }
```

2. 编写在 Frame 中显示不同大小不同风格、类型的字体。

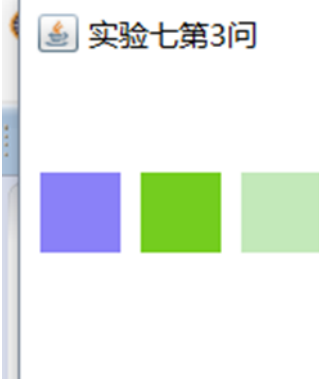
(运行截图+代码截图)



```
1 package exercise_7;
2 import java.awt.*;
3
4
5 public class test2 extends JApplet{
6
7     public void paint(Graphics g){
8         Font myfont1=new Font("Arial",Font.ITALIC,15);
9         Font myfont2=new Font("楷体",Font.BOLD,20);
10        Font myfont3=new Font("宋体",Font.PLAIN,25);
11        g.setFont(myfont1);
12        g.drawString("Font(Arial,Font.ITALIC,15);", 10, 20);
13        g.setFont(myfont2);
14        g.drawString("Font(楷体,Font.BOLD,20);", 10, 50);
15        g.setFont(myfont3);
16        g.drawString("Font(宋体,Font.PLAIN,25)", 10, 80);
17    }
18 }
19
20
21
```

3. 编写在 Panel 中显示一排用随机定义的颜色所填充的小方块。

(运行截图+代码截图)

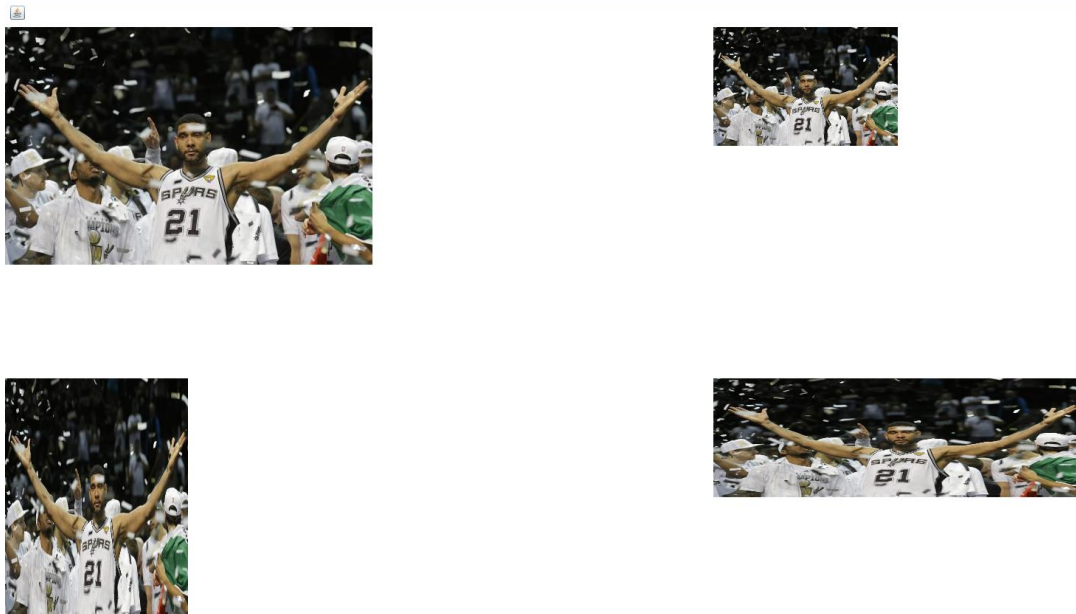


```
package exercise_7;
import java.awt.*;
import javax.swing.*;
import java.util.Random;
public class test3 extends JFrame{

    Random random=new Random();
    int red,green,blue;
    JPanel mypanel=new JPanel();

    test3(){
        this.getContentPane().add(mypanel);
    }
    public void getrandom_color(){
        red=random.nextInt(256);
        green=random.nextInt(256);
        blue=random.nextInt(256);
    }
    public void paint(Graphics g){
        g=mypanel.getGraphics();
        getrandom_color();
        Color color1=new Color(red,green,blue);
        g.setColor(color1);
        g.fillRect(10,50,40,40);
        getrandom_color();
        Color color2=new Color(red,green,blue);
        g.setColor(color2);
        g.fillRect(60,50,40,40);
        getrandom_color();
        Color color3=new Color(red,green,blue);
        g.setColor(color3);
        g.fillRect(110,50,40,40);
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        test3 my_frame=new test3();
        my_frame.setDefaultCloseOperation(3);
        my_frame.setTitle("实验七第3问");
        my_frame.setSize(200,200);
        my_frame.setVisible(true);
    }
}
```

-
4. 编写在 Panel 中显示图片文件，包括原图、缩小一半图、宽扁图和瘦高图。

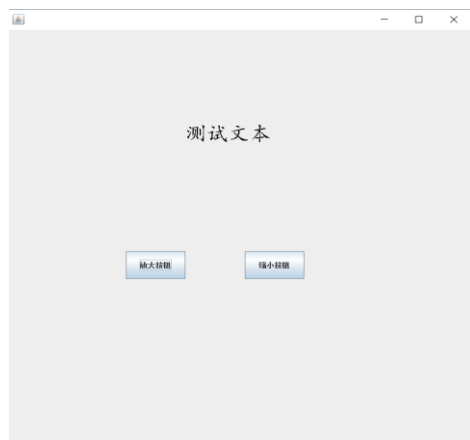


```
package exercise_7;
import java.awt.*;

public class test4 extends JFrame{
    private static final long serialVersionUID = 1L;
    JPanel jpanel1=new JPanel();
    JPanel jpanel2=new JPanel();
    JPanel jpanel3=new JPanel();
    JPanel jpanel4=new JPanel();
    GridLayout gridlayout1=new GridLayout(2,2);
    Image img=getToolkit().getImage("spurs.jpg");

    /* public int x=img.getWidth(this);
    public int y=img.getHeight(this);*/
    test4(){
        this.setLayout(gridlayout1);
        this.getContentPane().add(jpanel1);
        this.getContentPane().add(jpanel2);
        this.getContentPane().add(jpanel3);
        this.getContentPane().add(jpanel4);
    }
    public void paint(Graphics g){
        g=jpanel1.getGraphics();
        g.drawImage(img, 0, 0, this);
        g=jpanel2.getGraphics();
        g.drawImage(img, 0, 0, img.getWidth(this)/2, img.getHeight(this)/2, this);
        g=jpanel3.getGraphics();
        g.drawImage(img, 0, 0, img.getWidth(this)/2, img.getHeight(this), this);
        g=jpanel4.getGraphics();
        g.drawImage(img, 0, 0, img.getWidth(this), img.getHeight(this)/2, this);
    }
}
```

5. 编写在 Panel 中显示，包含两个按钮，一个按钮用于放大 Panel 上的一串字符串，一个按钮用于缩小；连续点击可不断放大或缩小。



```
package exercise_7;
import java.awt.*;

public class test5 extends JFrame{
    JPanel panell=new JPanel();
    JButton magnify=new JButton("放大按钮");
    JButton shrink=new JButton("缩小按钮");
    int Fontsize=30;
    Font myfont=new Font("楷体",Font.PLAIN,Fontsize);
    test5(){
        try{
            init();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
    public void init(){
        this.getContentPane().setLayout(null);
        panell.setBounds(0, 0, 800, 400);
        magnify.setBounds(200, 400, 100, 50);
        shrink.setBounds(400,400,100,50);
        this.getContentPane().add(panell);
        this.getContentPane().add(shrink);
        this.getContentPane().add(magnify);

        shrink.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                shrink_actionperformed();
            }
        });
        magnify.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                magnify_actionperformed();
            }
        });
    }
    public void shrink_actionperformed(){
        if(Fontsize>4) Fontsize--;
        myfont=new Font("楷体",Font.PLAIN,Fontsize);
        this.repaint();
    }
    public void magnify_actionperformed(){
        Fontsize++;
        myfont=new Font("楷体",Font.PLAIN,Fontsize);
        this.repaint();
    }
    public void paint(Graphics g){
        super.paint(g);
        g=panell.getGraphics();
        g.setFont(myfont);
        g.drawString("测试文本", 300, 200);
    }
}
```

三. 思考与分析

(1) Graphics 抽象类是如何实现图形绘制的?

在组件 Component 类中重写 paint() 方法, 可实现在组件上绘制图形的目的。
其实我也不是太懂这块。。。

解析3: public void paint(Graphics g)里面的g并非java.awt.Graphics, 而是Graphics context(图形语义环境)---我也不是很清楚这是什么东东, 但api里说到它的功能是允许应用程序在component上画东西显示在不同的设备上的一种环境。所有component的Graphics context都继承了java.awt.Graphics, applet也是Component的子类, 故这里的Graphics g 就不是abstract Graphics了, 而是Applet的Graphics context

来源百度

(2) 在 GUI 组件中文字是如何处理的?

GUI 中的文字也是以图形方式构成的, 可以用 Font 类设置需要显示文字的属性
(字体、字号等)

四、总结 (实验中遇到的问题、取得的经验、感想等)

在做第五问的时候, 重写 paint() 时候要添加 super.paint(g); 否则每一次 repaint() 都是在原有基础上绘制, 会出现文字重叠。