



中南大学
CENTRAL SOUTH UNIVERSITY

JAVA 实验报告

实验一 运用JavaFx画时钟

班 级： 计算机科学与技术 1306

学 号： 0902130610

姓 名： 王 雨 思

指导教师： 鲁 鸣 鸣

2014 年 12 月 1 日

目 录

一. 概述	6
二. 总体方案设计	7
三. 详细设计	8
四. 程序的调试与运行结果说明	9
五. 课程设计总结	10
六. 后记	11
七. 附录	12
参考文献	13

一 概述

1. 课程设计的目的

- 1、 设计一个带有指针、数字时钟，能方便我们的使用和操作，为我们以后的使用创造一个良好的平台。了解和掌握 JavaFx 基本概念和用法。
- 2、 把自己所学的理论知识运用到实际操作中去，以此来发现自己的不足，及时的改正。在实际的操作中学习及运用还没有学过的知识。最后在调试与测试的过程还可以提升自己的纠错能力也是自己经验的积累。

2. 课程设计的要求

编写一个模拟时钟程序，此程序在屏幕左方有一指针式钟面，右方有两个矩形框，上面以数字方式显示日期和时间，该时间应与指针显示的时间一致，下方的矩形框作为秒表。用菜单选项或按钮设置时间和秒表。时间不必与机器系统时间相同，只要可任意设置即可。

在设计的过程中，需要用到 `awt` 组件和 `Swing` 容器与组件，布局管理器，事件处理，绘制图形，还运用到了类的继承，事件监听器和线程。还有自己还没有掌握的 2D 绘制图形，以及绘制各指针及之间的关系。为了完成自己的设计，必须 API、教材以及相关的资料来解决这些尚未掌握的问题。

老师的要求就是尽量实现设计的功能，尽可能的锻炼自己的动手能力，提高自己的基础知识。

3. 课程设计的主要设计思想

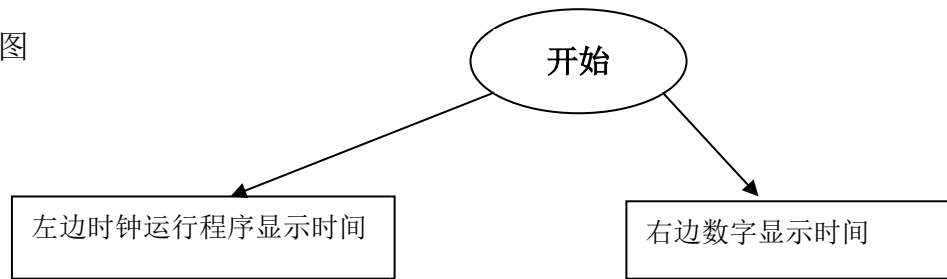
先是利用构造函数创建一个长为 500，宽为 200 的窗体，颜色设置为 `black`，并把窗体设计为固定大小，并且创建一个监听事件进行监听。随即创建一个时间计数器，最后实现 `ActionListener` 接口的实现方法。利用 2D 绘制图形的方法，在窗体左方画一个圆，并且绘制时钟上的 12 个数字刻度，然后利用 `Calendar` 类提供的 `HOUR`，`MINUTE`，`SECOND` 和 `MILLISECOND` 方法获得现在的时间，在计算时间和度数的关系；在利用 2D 绘制图形的方法绘制

出各指针，并设置不同的颜色。在利用 `drawString` 在窗体的右上方画出一个小的框并在里面现在和时钟相对数字时间。

二 总体方案设计

- 1、 编写一个模拟时钟程序，此程序在屏幕左方有一指针式钟面。
- 2、 右方有矩形框，上面以数字方式显示日期和时间，该时间应与指针显示的时间一致。

整体框图



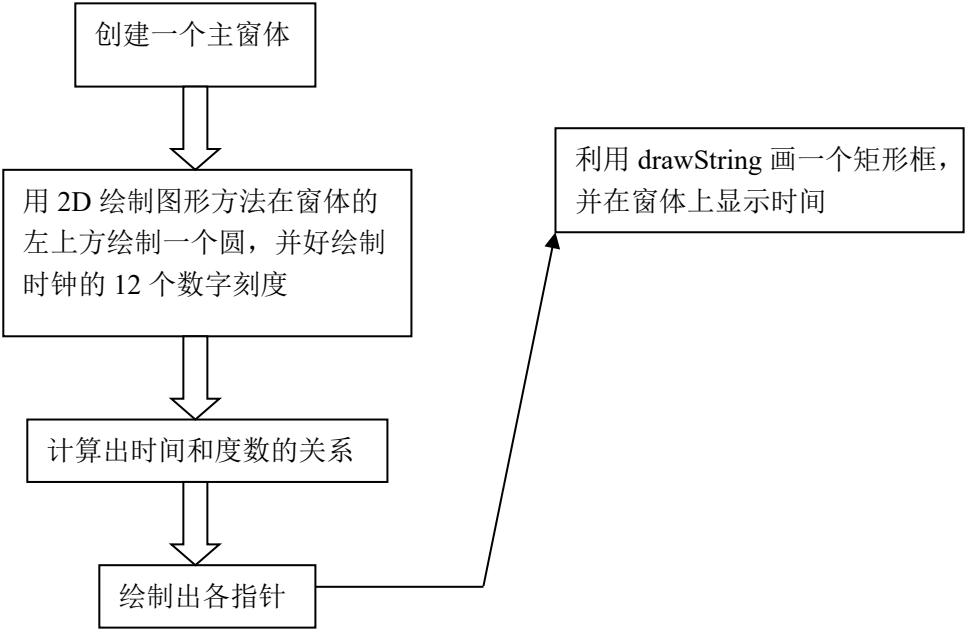
三 详细设计

- 1、 先是利用构造函数创建一个长为 400，宽为 400 的窗体，颜色设置为 `white`，并把窗体设计为固定大小，并且创建一个监听事件进行监听。随即创建一个时间计数器，最后实现 `ActionListener` 接口的实现方法。
- 2、 利用 2D 绘制图形的方法，在窗体左方画一个圆，并且绘制时钟上的 12 个汉字，然后利用 `Calendar` 类提供的 `HOUR`，`MINUTE`，`SECOND` 和 `MILLISECOND` 方法获得现在的时间，在计算时间和度数的关系；在利用 2D 绘制图形的方法绘制出各指针，并设置不同的颜色。
- 3、 在利用 `drawString` 在窗体的右上方画出一个小的框并在里面现在和时钟相对数字时间。
- 4、 在编写主函数，在运行程序。

所完成的具体功能及用到的算法（详细分析）。

程序流程图

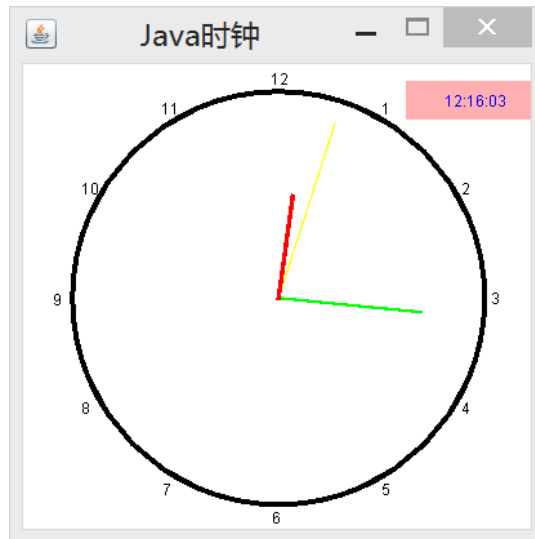
主要部分的详细流程图



四 程序的调试与运行结果说明

调试的方法：首先我是一个类一个类的完成，在每一个类完成后都会对它进行编译，待它能正常的编译之后在进行下面的类的编写。在编译的过程中会有一些错误，比如说语法错误，类的引用格式不正确，会产生一些异常。但是在经过慢慢的调试，这些问题逐渐得到解决。

运行结果：



五 课程设计总结

对于课程设计，基本上实现了模拟时钟程序的要求，此程序在屏幕左方有一指针式钟面，右方有矩形框，上面以数字方式显示日期和时间，该时间应与指针显示的时间一致。

该设计的特点是时钟上标有 12 个刻度，并且各指针是不同的颜色，看上去比较美观，在右上方有两个矩形框，第一个矩形框显示数字时间，这是为了能更好的看时间。

我想在不断地进行对该设计的完善，不仅能为该时钟设置背景，以增加该时钟的美观效果，也可以对时钟到了一定的时间就可以报时，并且可以设置闹钟，为闹钟设置多种音乐，可供选择，这样的话就不显得单调，这样一来该时钟更方便实用。并且加上开始停止按钮。

困难及解决途径

在上机实习完成课程设计的过程中，遇到了不少的问题，一是有一些东西自己没有学过，而是在编写的过程中由于思路不清晰以及自己的粗心给自己制造了一些麻烦，还有就是会产生一些异常。但是在经过自己和老师、同学的交流，以及自己查阅资料和 API 文档，针对性的对不懂得问题进行查询，自己没有学过但是又要用到得逐渐的得到解决，再经过自己反复的理清思路和反复的检查由思路不清晰和粗心带来的麻烦也慢慢得到了解决。

六 后记

首先，java 作为一种计算机语言，它具备语言的特性，就是我们在课堂上认真听取了老师的指导之后，还得自己回去认真看书写程序，并且上机实习才

能发现问题及时解决问题。就像我们学习语言一样要通过和同学之间的交流才能提高我们的实际应用能力。所以，我们在自己认真刻苦学习的同时，一定要多和同学老师交流，从他们那里吸取好的思维方式、好的学习方法。

在上机实习的时候，我现在回顾自己一学期的学习，觉得首先应该熟悉 java 的概念，明白程序开发的基本思想，熟悉开发环境之后马上进行自主学习，自己学习的步伐千万不能落于老师讲课授课的速度。如果这样的话，我们很可能处于一种被动学习的劣势，失去对 java 学习的热情，以至于恶性循环，到最终的无可救药。

在平时的训练和刚刚结束的上机实习中我深刻的体会到了学习好 java 我们就要一定要掌握好 java 的基本语句，包括条件语句和循环语句。这些就是我们编写程序这棵大树的必要组成枝叶。建立起关键语句的之后，我们才能有更强的功能和实用性。

Java 作为一个面向对象的编程语言，在图像方面具有很强的实现能力。这样很能激发我们的学习兴趣。在编写程序的时候发现，虽然它是一个面向对象的编程语言，我们一定要学会如何编写多个类和方法，在不断地摸索中培养我们自身设计面向对象解决问题的能力。作为学生的我们，是一个学习的过程，我们应该学习如何写好程序。而好的程序不仅仅是一系列语句的拼凑，我们要联系自身实际，用编程语言解决问题和实施解决问题的方案。在学习 java 的这段时间当中，图形激发了我学习的巨大热情。所以，我觉得我自己一定要在打好坚实的基础的情况下，对图形学进行认真学习。在学习图形和图形用户接口的时候，觉得变得吃力起来。因为随着课程的加重，发现自己动手能力越来越若，在事件处理和图形用户接口的使用中，出现了知识点模棱两可的情况。当然，任何的困难挫折在不断地努力中都会变得脆弱，只是暂时的阴影。在以后的日子里，我要不断地学习这个博大精深的语言，继续不断的深化自己的实际动手能力。

七 附录

```
package clock1;
```

```

import javax.swing.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.*;
import java.util.Calendar;
import java.util.GregorianCalendar;

class ClockPointer extends JFrame implements ActionListener{

    int
    x,y,x0,y0,r,h,olds_x,olds_y,oldm_x,oldm_y,oldh_x,oldh_y,ss,mm,hh,old_m,old_
    h,ang;
    final double RAD=Math.PI/180;

    //构造函数创建一个窗体
    public ClockPointer(){
        super("Java 时钟");
        setDefaultCloseOperation(3);
        setSize(400,400);
        setBackground(Color.WHITE);
        setLocation(300,150);
        setResizable(false);
        setVisible(true);
        int delay = 1000;
        //创建一个监听事件
        ActionListener drawTestClock = new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                repaint();
            }
        };
        //创建一个时间计数器
        new Timer(delay,drawTestClock).start();
    }
    //实现 ActionListener 接口必须实现的方法
    public void actionPerformed(ActionEvent e){
    }

    //绘制图形
    public void paint(Graphics g){
        Graphics2D g2D = (Graphics2D)g;
        Insets insets = getInsets();
        int L = insets.left/2,T = insets.top/2;
        h = getSize().height;
        g.setColor(Color.black);
    }

```



```

//画圆
g2D.setStroke(new BasicStroke(4.0f));
g.drawOval(L+40,T+40,h-80,h-80);
r=h/2-40;
x0=40+r-5+L;
y0=40+r-5-T;
ang=60;
//绘制时钟上的12个汉字
for(int i=1;i<=12;i++){
    x=(int)((r+10)*Math.cos(RAD*ang)+x0);
    y=(int)((r+10)*Math.sin(RAD*ang)+y0);
    g.drawString(""+i,x,h-y);
    ang-=30;
}
//获得现在时间

Calendar now= new GregorianCalendar();
int nowh= now.get(Calendar.HOUR_OF_DAY);
int nowm= now.get(Calendar.MINUTE);
int nows= now.get(Calendar.SECOND);
int nowmi= now.get(Calendar.MILLISECOND);
String st;
if(nowh<10) st="0"+nowh;else st=""+nowh;
if(nowm<10) st+=":0"+nowm;else st+=":"+nowm;
if(nows<10) st+=":0"+nows;else st+=":"+nows;

//在窗体上显示时间
g.setColor(Color.pink);
g.fillRect(300,50,100,30);
g.setColor(Color.blue);
g.drawString(st,330,70);

```

```

//计算时间与度数的关系
ss=90-nows*6;
mm=90-nowm*6;
hh=90-nowh*30-nowm/2;
x0=r+40+L;
y0=r+40+T;
g2D.setStroke(new BasicStroke(1.2f));
//擦除秒针
if(olds_x>0){

```

```

        g.setColor(getBackground());
        g.drawLine(x0,y0,olds_x,h-olds_y);
    }
    else{
        old_m = mm;
        old_h = hh;
    }
    //绘制秒针
    x=(int)(r*0.9*Math.cos(RAD*ss))+x0;
    y=(int)(r*0.9*Math.sin(RAD*ss))+y0-2*T;
    g.setColor(Color.yellow);
    g.drawLine(x0,y0,x,h-y);
    olds_x=x;
    olds_y=y;
    g2D.setStroke(new BasicStroke(2.2f));
    //擦除分针
    if(old_m!=mm){
        g.setColor(getBackground());
        g.drawLine(x0,y0,oldm_x,h-oldm_y);
    }
    //绘制分针
    x=(int)(r*0.7*Math.cos(RAD*mm))+x0;
    y=(int)(r*0.7*Math.sin(RAD*mm))+y0-2*T;
    g.setColor(Color.green);
    g.drawLine(x0,y0,x,h-y);
    oldm_x=x;
    oldm_y=y;
    old_m=mm;
    g2D.setStroke(new BasicStroke(3.4f));
    //擦除时针
    if(old_h!=hh){
        g.setColor(getBackground());
        g.drawLine(x0,y0,oldh_x,h-oldh_y);
    }
    //绘制时针
    x=(int)(r*0.5*Math.cos(RAD*hh))+x0;
    y=(int)(r*0.5*Math.sin(RAD*hh))+y0-2*T;
    g.setColor(Color.red);
    g.drawLine(x0,y0,x,h-y);
    oldh_x=x;
    oldh_y=y;
    old_h=hh;
}

```

```
public static void main(String[] args){  
    new ClockPointer();  
}
```

```
public static void main(String[] args){  
    new TestClock();  
}
```

参考文献

[1] Java 语言程序设计 郭克华 段桂华 清华大学出版社