报告一：Window 的时钟和日历动画

第1部分　设计的内容要求

1.1 设计内容

模仿 Window 的时钟和日历动画，设计该应用程序

1.2 设计要求

1. 实现 Windows 中时钟日历应用程序的所有功能
2. 鼓励添加更加新颖的特色功能

第2部分　设计思想

2.1 主要思想

主要利用 Runnable 实现多线程动画程序的编写，主框架里添加两个面板，分别用以实现日历程序的实现和钟表程序的实现。

2.2 算法

日历功能主要使用了 Java 自带的 calendar 类，用以获取当前时间以及星期，将面板利用 GridLayout 布局分成 7x7 的布局，填充 label 组件，再编写函数来根据用户所选年月进行 label 文本的设置，从而达到日历的显示功能，分别在年份和月份两旁添加 button 及监听器，用以实现月份或者年份的选择，再传至函数进行判断。

时钟功能主要使用了 Runnable 进行多线程的实现，单独创建了一个继承 Panel 的类来画时钟以及同步日期，时间的显示，其中利用 sleep 函数暂停一秒，达到一秒刷新一次的效果，时钟主要利用定坐标画线条的方法，利用不断重画，将上一次的指针线条颜色与背景色相同达到消除的效果。

第3部分　详细设计

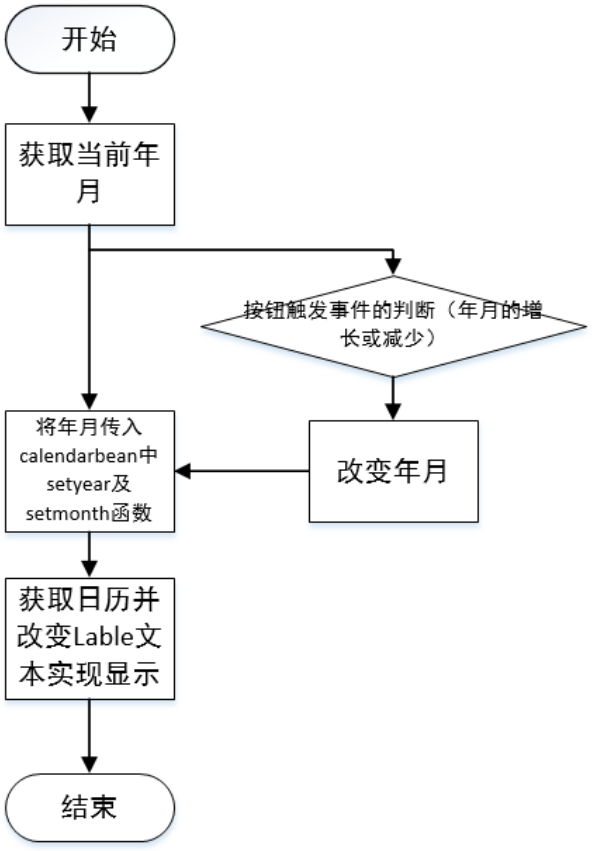
3.1 日历功能

1. 详细实现过程

首先新写了一个继承 Panel 的类，使用 BorderLayout 和 GridLayout 布局管理器结合布局，大体分为年月的操作及显示面板和具体的星期与日期的显示面板，在年月的操作及显示面板用 Label 组件实现显示，Button 组件实现操作，操作分为对年份的增减和对月份的增减，为 Button 组件注册监听器，实现每点击一次，改变 Label 组件的文本以及重新获取星期与日期显示面板。

为了星期与日期显示面板的实现，重新创建了一个类 CalendarBean 用以判断年月日，同时获取当月一日的具体星期，在类中可以生成一个排列好的 42 位的字符串数组，用以更改填充日历面板的 Label 的文本，达到日历的显示效果。

2. 流程图

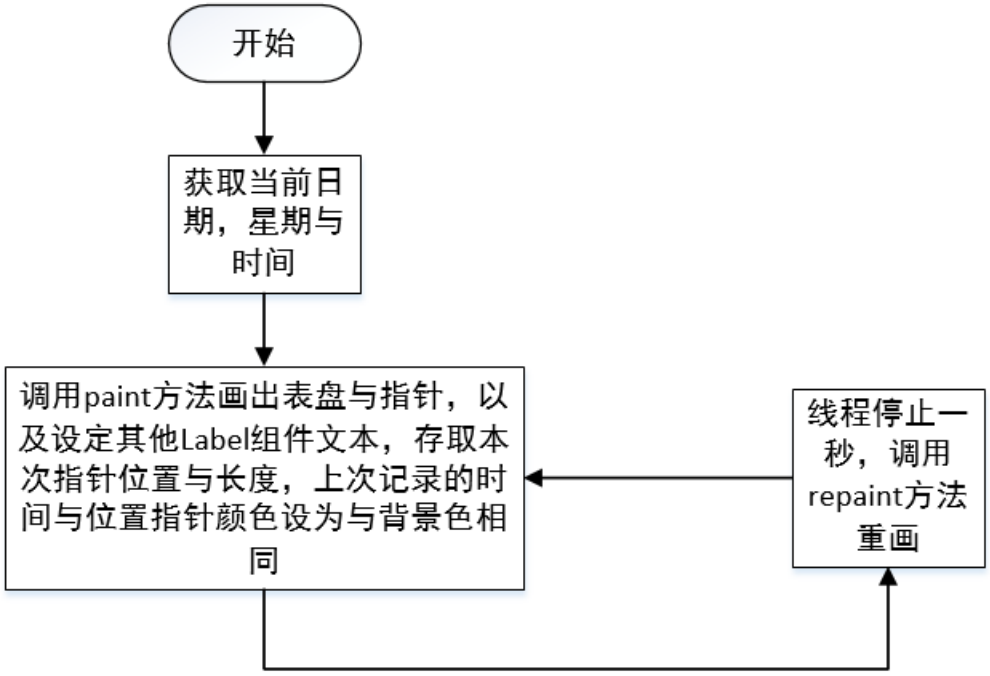


3.2 钟表功能

1. 详细实现过程

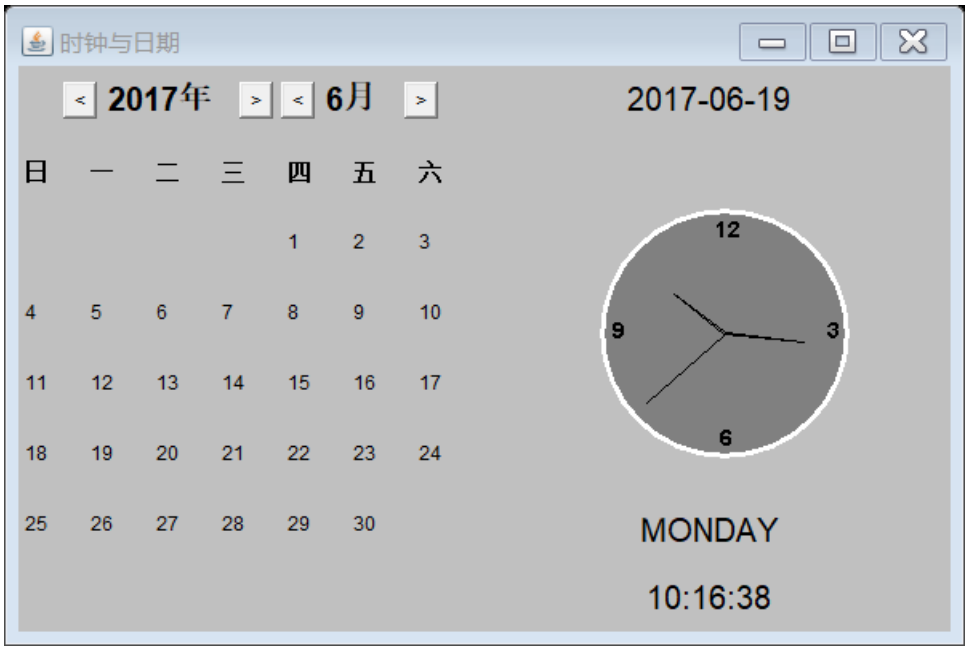
在钟表面板里具体有四个显示面板，利用 GridLayout 布局管理器分开，分别用以显示日期，钟表，星期，具体时间。利用接口 Runnable，其中 paint 函数实现了表盘的绘制以及其他三条文本的显示与刷新，在 run 方法中利用 sleep 方法暂停一秒再调用 repaint 方法重画以达到一秒刷新一次的效果，钟表在刷新的时候会根据当前系统时间重新画表盘与指针同时存下这次指针的位置及长度，之前的指针颜色改为表盘颜色，这样就可以达到指针在转动的动画效果，同时每次刷新时，对显示日期，星期，具体时间的 Label 文本进行重设，达到同步更新的效果。重写 update 方法达到消抖的效果。

2. 流程图

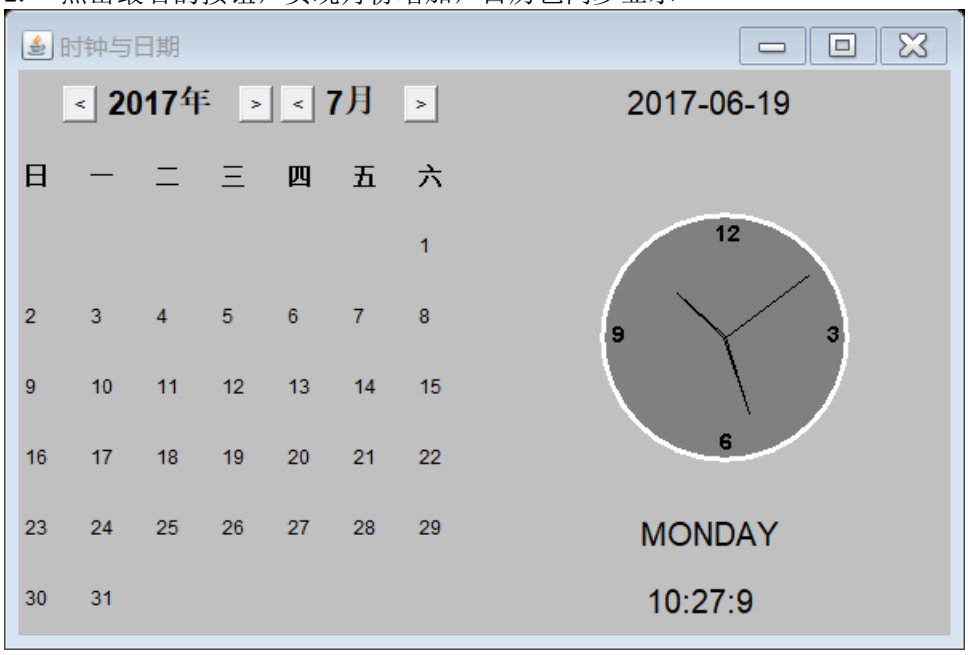


第4部分　运行效果图

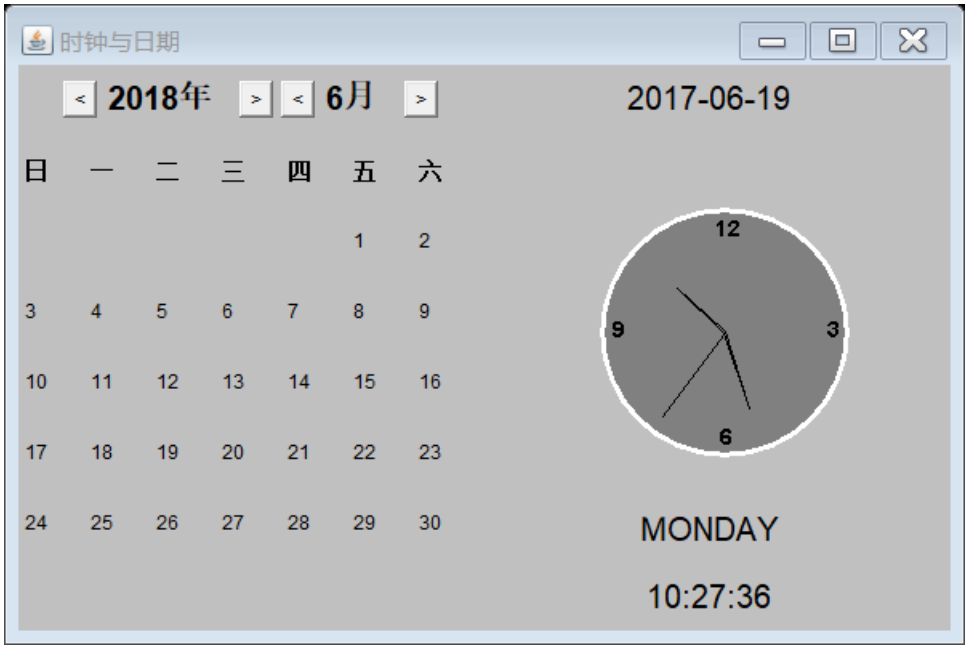
1. 主界面运行截图



2. 点击最右的按钮，实现月份增加，日历也同步显示



3. 点击第二个按钮，实现年份的增加。



第5部分　心得与体会

最开始做这个课设的时候，习惯性的用了 JButton,JLabel 等 Swing 里面的组件，却出现了很多意想不到的错误，在线程的进行中，JButton，JLabel 组件都无法直接显示，只有被选中才会出现，而 JLabel 组件无法选中导致无法显示，但最后将组件换为 Button，Label 问题就迎刃而解了。还出现过一个问题在文字一开始无法显示，需要人为的操作窗口以后才会出现，为了解决这个问题，在面板的初始化上就先加上了显示当前信息的 Label，从而在后来随着线程的运行而不断刷新。同时，较之于之前作业中获取时间的方法，在新版的 Java 中方法已经不被推荐，故上网查询了 LocalDate 等类的用法，还有在日历的编写中，也学习了 Calendar 类的使用方法。总之，通过这个课设，对于线程的理解也要更近一步，更深刻的理解了 run 方法，paint 方法，repaint 方法，update 方法在多线程编程中的运用。

第6部分　附录

```
//钟表的显示类核心代码
class Cpanel extends Panel implements Runnable{

	public void paint(Graphics g)  //显示数字和图形时钟的方法
	{
		int xh, yh, xm, ym, xs, ys, s, m, h, xcenter, ycenter;
		LocalTime time=LocalTime.now();
		LocalDate date=LocalDate.now();
		s = time.getSecond();
		m = time.getMinute();
		h = time.getHour();
		l1.setText(h+":"+m+":"+s);
		l2.setText(date.toString());
		l3.setText((date.getDayOfWeek()).toString());
		xcenter=75+75; ycenter=50+75; //图形钟的原点
		//以下计算秒针、分针、时针位置
		xs = (int)(Math.cos(s * 3.14f/30 - 3.14f/2) * 65 + xcenter);
		ys = (int)(Math.sin(s * 3.14f/30 - 3.14f/2) * 65 + ycenter);
		xm = (int)(Math.cos(m * 3.14f/30 - 3.14f/2) * 50 + xcenter);
		ym = (int)(Math.sin(m * 3.14f/30 - 3.14f/2) * 50 + ycenter);
		xh = (int)(Math.cos((h*30+m/2)*3.14f/180-3.14f/2)*40+xcenter);
		yh = (int)(Math.sin((h*30+m/2)*3.14f/180-3.14f/2)*40+ycenter);
		g.setFont(new Font("TimesRoman", Font.BOLD, 14));
		g.setColor(Color.WHITE); //设置表盘颜色
		g.fillOval(xcenter-78,ycenter-78,156,156); //画表盘
		g.setColor(Color.GRAY); //设置表盘颜色
		g.fillOval(xcenter-75,ycenter-75,150,150); //画表盘

		g.setColor(Color.black); //设置表盘数字颜色
		g.drawString("9",xcenter-70,ycenter+3); //画表盘上的数字
		g.drawString("3",xcenter+64,ycenter+3);
		g.drawString("12",xcenter-6,ycenter-60);
		g.drawString("6",xcenter-3,ycenter+70);
		//时间变化时，需要重新画各个指针，即先消除原有指针，然后画新指针
		g.setColor(Color.GRAY); //用表的填充色画线，可以消除原来画的线
		if (xs != lastxs || ys != lastys){ //秒针变化
			g.drawLine(xcenter, ycenter, lastxs, lastys); }
		if (xm != lastxm || ym != lastym) { //分针变化
			g.drawLine(xcenter, ycenter-1, lastxm, lastym);
			g.drawLine(xcenter-1, ycenter, lastxm, lastym); }
		if (xh != lastxh || yh != lastyh) { //时针变化
			g.drawLine(xcenter, ycenter-1, lastxh, lastyh);
			g.drawLine(xcenter-1, ycenter, lastxh, lastyh); }
		g.setColor(Color.black); //画新指针
		g.drawLine(xcenter, ycenter, xs, ys);
		g.drawLine(xcenter, ycenter-1, xm, ym);
		g.drawLine(xcenter-1, ycenter, xm, ym);
		g.drawLine(xcenter, ycenter-1, xh, yh);
		g.drawLine(xcenter-1, ycenter, xh, yh);
		lastxs=xs; lastys=ys; //保存指针位置
		lastxm=xm; lastym=ym;
		lastxh=xh; lastyh=yh;
```

```
	}
	public void run(){ //每隔一秒钟，刷新一次画面的方法
		while (timer != null)
		{
			try { Thread.sleep(1000); }
			catch (InterruptedException e) {}
			repaint(); //调用paint()方法重画时钟
		}
		timer = null;
	}
	public void update(Graphics g) //重写该方法是为了消除抖动现象
	{
		paint(g);
	}
}
//日历的显示类核心代码
class Calen extends Panel implements ActionListener
{
	public Calen()
	{
		int i ;
		b1=new Button("<");b2=new Button(">");
		b3=new Button("<");b4=new Button(">");
		b1.addActionListener(this);b2.addActionListener(this);
		b3.addActionListener(this);b4.addActionListener(this);
		calendar=new CalendarBean();
		calendar.setYear(year);
		calendar.setMonth(month);
		String []day=calendar.getCalendar();
		l1.setText(String.valueOf(date.getYear())+"年");
		l2.setText(String.valueOf(date.getMonthValue())+"月");
		l1.setFont(new Font("TimesRoman", Font.BOLD, 20));
		l2.setFont(new Font("TimesRoman", Font.BOLD, 20));
		this.setLayout(new BorderLayout());
		this.add(p1,"North");this.add(p2);
		p1.add(b3);p1.add(l1);p1.add(b4);p1.add(b1);p1.add(l2);p1.add(b2);
		p2.setLayout(new GridLayout(7,7));
		try{for( i = 0; i<7;i++)
		{
			this.week[i]=new Label("");
			this.week[i].setText(week_name[i]);
			this.week[i].setFont(new Font("TimesRoman", Font.BOLD, 14));
		}
		for( i = 0; i<7;i++)
		{
			p2.add(week[i]);
		}
		for(i=0;i<42;i++)
		{
			day_list[i]=new Label();
			day_list[i].setText(day[i]);
		}
		for(i=7;i<49;i++)
		{p2.add(day_list[i-7]);}
		}catch(Exception e){
		}
	}
```

```
	@Override
	public void actionPerformed(ActionEvent e) {
		if(e.getSource()==b2)
		{	month=month+1;
			if(month>12)
			{year=year+1;
			month=1;
			}
			calendar.setYear(year);
			calendar.setMonth(month);
			String day[]=calendar.getCalendar();
			for(int i=0;i<42;i++)
			{day_list[i].setText(day[i]);}
			l1.setText(String.valueOf(year)+"年");
			l2.setText(String.valueOf(month)+"月");
			l1.setFont(new Font("TimesRoman", Font.BOLD, 20));
			l2.setFont(new Font("TimesRoman", Font.BOLD, 20));
		}
		//监听器触发事件其他代码类似，故省略。
	}
}
//日历的判断类核心代码
class CalendarBean
{
	public String[] getCalendar()
	{

		//判断月份
		if(month==1||month==3||month==5||month==7
				||month==8||month==10||month==12)
		{	day=31;}
		if(month==4||month==6||month==9||month==11)
		{	day=30;}
		//判断平年与闰年
		if(month==2)
		{if(((year%4==0)&&(year%100!=0))||(year%400==0))
			{	day=29;}
			else
			{	day=28;}
		}
		for(int i=week,n=1;i<week+day;i++)
		{
			a[i]=String.valueOf(n) ;
			n++;
		}
		return a;
	}
}
```