单元 2.5.2 跑步比赛（二）

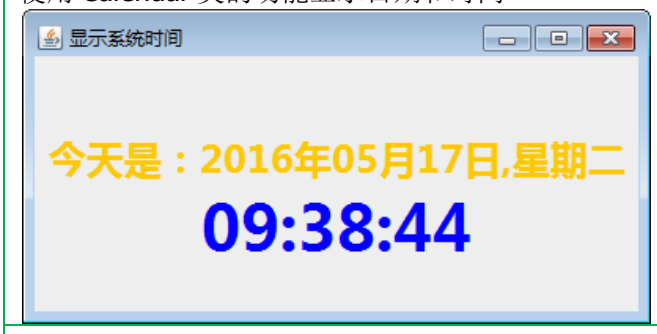
单元教学设计

<table>
<tr><td colspan="2">学习阶段</td><td colspan="2">二、实用程序开发训练</td><td colspan="2">学时</td></tr>
<tr><td colspan="2">项目 2.5</td><td colspan="2">跑步比赛</td><td>学时</td><td>6</td></tr>
<tr><td colspan="2">单元 2.5.2</td><td colspan="2">跑步比赛（一）</td><td>学时</td><td>2</td></tr>
<tr><td rowspan="3">教学目标</td><td>能力目标</td><td colspan="2">知识目标</td><td colspan="2">思政目标</td></tr>
<tr><td rowspan="2">能够理解线程的概念，并用线程来解决实际问题。
能够运用多媒体，播放背景音乐。</td><td>了解线程的四种状态
了解线程的优先级及优先级的设定</td><td>课前知识储备</td><td rowspan="2">在东京奥运会上，每一位运动员都值得被尊重，当然比赛就会有输赢，不要感到气馁，继续加油，争取在下一次做到更好。只有打开自己的潜能，才能勇敢的面对未来，才能无惧一切未知。
永不放弃，更是一种高宝贵的品质。在成功的路上，一定会遇到许多许许多多困难，只有那些不畏艰险，执着追求的人才有希望达到顶点。。</td></tr>
<tr><td>1.掌握 Java 播放多媒体的方法
2.掌握用线程计时的方法。
3.熟练掌握线程结束的方法
4.熟练掌握日期类</td><td>课上学习练习</td></tr>
<tr><td rowspan="3">课前准备</td><td>视频</td><td colspan="3">讲座：《Java 中的时间处理》</td></tr>
<tr><td>课前练习</td><td colspan="3">获得系统时间，并以如下格式显示：</td></tr>
<tr><td>十分钟测试</td><td colspan="3">使用 Calendar 类的功能显示日期和时间</td></tr>
<tr><td>新课导入</td><td colspan="4">在上节课中，我们实现了 Thread 公司的代言人小破孩和 Runnable 公司代言人小丫的跑步比赛，为了让比赛更加激情澎湃，我们可以在赛跑开始的时候加上比赛开始的枪声，在比赛过程中，加上观众席的“加油”声，这样，比赛将会更加精彩。
另外，为了精确比较位运动员的比赛成绩，我们可以记录下来两位选手到达重点时候的时间，这样会让比赛更加公平。</td></tr>
<tr><td>项目进度安排</td><td colspan="4"><table><tr><th>任务</th><th>功能要求</th><th>训练技能点</th><th>完成方式</th></tr><tr><td>1</td><td>添加“开始”按钮，点击按钮后比赛开始</td><td>线程的调用</td><td>教师：描述需求
学生：编程实现</td></tr><tr><td>2</td><td>比赛开始时，倒计时 3.2.1 后，枪声响起，同时运动员开始比赛</td><td>播放声音</td><td>教师：演示讲解
学生：编程实现</td></tr><tr><td>3</td><td>比赛过程中，观众席加油，直到最后一名运动员到达终点</td><td>线程的终止</td><td>教师：演示讲解
学生：编程实现</td></tr><tr><td>4</td><td>比赛过程中，显示秒表计时，比赛结束后，显示两位选手的成绩。</td><td>时间和时间类</td><td>教师：演示讲解
学生：编程实现</td></tr></table></td></tr>
<tr><td>十分钟测试</td><td colspan="4">2.练习：
（1）做窗口 1，上有一个 JtextArea，设置为不可编辑，有一个 Jbutton。
（2）点击窗口 1 的 Jbutton，打开窗口 2
（3）窗口 2 上有输入框 JtextField，和按钮，在输入框中输入文本后，点击按钮，将输入框中的语句追加到窗口 1 的 JtextArea 上。</td></tr>
<tr><td rowspan="2">能力训练任务</td><td>任务一</td><td colspan="3">添加“开始”按钮，点击按钮后比赛开始
运行上节课的程序，发现程序一运行，两个小孩就开始跑，这点不符合比赛规则，为了更加实际，我们添加“开始”按钮，点击开始按钮后，两个小人再开始赛跑。</td></tr>
<tr><td>任务二</td><td colspan="3">点击“开始”按钮，比赛开始后枪声响起，同时运动员开始赛跑。
课程思政
知识点：线程调度算法
思政点：遵守规则，为自己创造机会
思政融入：1.在道路上驾驶机动车一定要遵守交通规则，注意交通安全！道路千万条，安全第一条！</td></tr>
</table>

		2．在遵守规则的基础上，通过努力增强自身的竞争力，在机会来到时会有更大的把握。
	任务三	比赛过程中，观众席加油，直到最后一名运动员到达终点，加油声停止。
	任务四	比赛过程中，显示秒表计时，比赛结束后，显示两位选手的成绩。
课后作业		课外同步项目
课后学习资源		在线学习平台
课程思政：		思政资源（微视频，课程平台）： 1.苏炳添究竟有多强？100 米跑进十秒究竟有多难？ 2.苏炳添奥运会创造历史，外国网友也被惊到：中国人能跑这么快！ 3.你是我的眼睛，我是你的骄傲！ 4.回顾苏炳添，全运会夺金，32 岁老将跑进十秒有多难？ 5.2020 东京奥运会：百米赛跑，中国选手一鸣惊人打破世界纪录！ 6.第 87 金！100 米 T36 级决赛 邓培程破残奥会纪录夺冠 思政主题： 1.了解跑步运动员是如何训练的？ 2.了解跑步竞赛要付出什么？

单元教学进度设计

Step1：项目导入（20）分钟

教学环节	教学内容	教师活动	学生活动
十分钟测试	使用 Calendar 类的功能显示日期和时间 	提问解析	抢答

	public class TestDate **extends** JFrame{
	JLabel l1=**new** JLabel();
	JLabel l2=**new** JLabel();
	Font font1=**new** Font("微软雅黑",Font.*BOLD*,24);
	Font font2=**new** Font("微软雅黑",Font.*BOLD*,40);
	public TestDate(){
	Container c=**this**.getContentPane();
	c.setLayout(**null**);
	GridLayout gl=**new** GridLayout(2,1);
	c.setLayout(gl);
	SimpleDateFormat format1=**new** SimpleDateFormat("今天
		辅导	编程

	是：yyyy年MM月dd日,E");
	SimpleDateFormat format2=**new** SimpleDateFormat("HH:mm:ss");
	Calendar rightNow = Calendar.*getInstance*();
	l1.setText(format1.format(rightNow.getTime()));
	l2.setText(format2.format(rightNow.getTime()));
	l1.setFont(font1);
	l1.setForeground(Color.*ORANGE*);
	l1.setVerticalAlignment(JLabel.*BOTTOM*);
	l1.setHorizontalAlignment(JLabel.*CENTER*);
	l2.setFont(font2);
	l2.setForeground(Color.*BLUE*);
	l2.setVerticalAlignment(JLabel.*TOP*);
	l2.setHorizontalAlignment(JLabel.*CENTER*);
	c.add(l1);
	c.add(l2);
	
	this.setSize(400, 200);
	this.setDefaultCloseOperation(JFrame.*EXIT_ON_CLOSE*);
	this.setTitle("显示系统时间");
	this.setVisible(**true**);
	}
	
	public static void main(String[] args){
	TestDate td=**new** TestDate();
	}
	}		

小结	**1. SimpleDateFormat** 在 Java 应用中，格式化日期时间通常会用到 SimpleDateFormat 类 public class SimpleDateFormat extends DateFormat SimpleDateFormat 是一个以国别敏感的方式格式化和分析数据的具体类。它允许格式化 (date -> text)、语法分析 (text -> date)和标准化。 G 年代标志符 y 年 M 月 d 日 h 时 在上午或下午 (1~12) H 时 在一天中 (0~23) m 分 s 秒 S 毫秒 E 星期 D 一年中的第几天	总结	记录

F	一月中第几个星期几
w	一年中第几个星期
W	一月中第几个星期
a	上午 / 下午 标记符
k	时 在一天中 (1~24)
K	时 在上午或下午 (0~11)
z	时区

2. Calendar 类能够支持不同的日历系统，它提供了多数日历系统所具有的一般功能，它是抽象类，这些功能对子类可用。
下边我们简要介绍一下 Calendar 类。

（1）类常数

该类提供了如下一些日常使用的静态数据成员：

1) AM（上午）、 PM（下午）、AM_PM（上午或下午）；
2) MONDAY~ SUNDAY（星期一 ~ 星期天）；
3) JANUARY ~ DECENBER（一月 ~ 十二月）；
4) ERA（公元或公元前）、YEAR（年）、MONTH（月） 、DATE（日）；
5) HOUR（时）、MINUTE（分）、SECOND（秒）、MILLISECOND（毫秒）；
6) WEEK_OF_MONTH（月中的第几周）、 WEEK_OF_YEAR（年中的第几周）；
7) DAY_OF_MONTH（当月的第几天）、DAY_OF_WEEK（星期几）、DAY_OD_YEAR（一年中第几天）等等。

另外还提供了一些受保护的数据成员，需要时请参阅文档。

（2）构造器

1) protected **Calendar**() 以系统默认的时区构建 Calendar。
2) protected **Calendar**(TimeZone zone, Locale aLocale) 以指定的时区构建 Calendar。

（3）常用方法

1) boolean **after**(Object when) 测试日期对象是否在对象 when 表示的日期之后。
2) boolean **before**(Object when) 测试日期对象是否在对象 when 表示的日期之前。
3) final void **set**(int year,int month,int date) 设置年、月、日。
4) final void **set**(int year,int month,int date,int hour,int minute,int second) 设置年、月、日、时、分、秒。
5) final void **setTime**(Date date) 以给出的日期设置时间。
6) public int **get(int field)** 返回给定日历字段的值。
7) static Calendar **getInstance**() 用默认或指定的时区得到一个对象。
8) final Date **getTime**() 获得表示时间值（毫秒）的 Date 对象。
9) TimeZone **getTimeZone**() 获得时区对象。
10) long **getTimeInMillis**() 返回该 Calendar 以毫秒计的时间。
11) static Calendar **getInstance**(Locale aLocale) 以指定的地点及默认的时区获得一个 calendar 对象。
12) static Calendar **getInstance**(TimeZone zone) 以指定的时区获得一个 calendar 对象。
13) static Calendar **getInstance**(TimeZone zone, Locale aLocale) 以指定的地点及时区获得一个 calendar 对象。

该类提供了丰富的处理日期和时间的方法，上边只是列出了一部份，详细内容请参阅 API 文档。

显示系统时间的基本步骤：

1.初始化 SimpleDateFormat 类的对象，定义时间显示的格式

SimpleDateFormat format1=new SimpleDateFormat("今天是：yyyy 年 MM 月

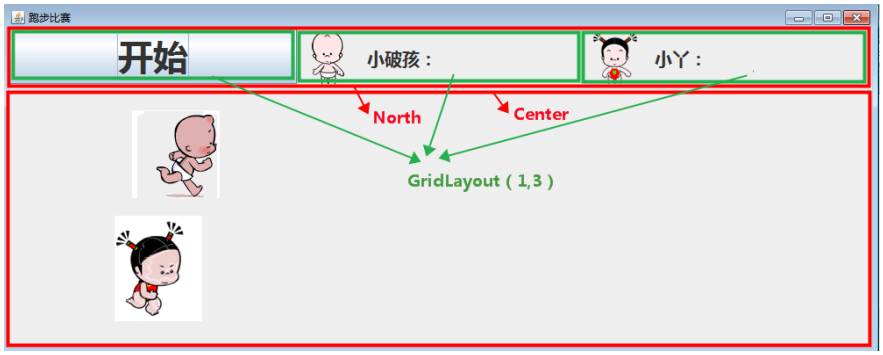
	dd 日,E"); 注意，在格式设置的时候，里面用特定的字符来标示时间，可以与汉字混排。 **2.初始化 Calendar 的对象。** Calendar rightNow = Calendar.getInstance(); 当然，也可以用 Date date =new Date( )，但是这种方式已经过时。 **3.获得系统时间** rightNow.getTime() 该方法返回的是一个 Date 类的对象。		

新课导入

在上节课中，我们实现了 Thread 公司的代言人小破孩和 Runnable 公司代言人小丫的跑步比赛，为了让比赛更加激情澎湃，我们可以在赛跑开始的时候加上比赛开始的枪声，在比赛过程中，加上观众席的“加油”声，这样，比赛将会更加精彩。 另外，为了精确比较位运动员的比赛成绩，我们可以记录下来两位选手到达重点时候的时间，这样会让比赛更加公平。	提出问题 分析	讨论 思考
演示程序最终运行结果	演示	观看 思考

Step2：任务实施

任务 1：（25 分钟）

任务引入	添加“开始”按钮，点击按钮后比赛开始 运行上节课的程序，发现程序一运行，两个小孩就开始跑，这点不符合比赛规则，为了更加实际，我们添加“开始”按钮，点击开始按钮后，两个小人再开始赛跑。我们可以用布局管理器来实现。 	布置任务	思考
任务实施	学生自行完成 1.注意灵活使用布局管理器 2.注意使用 Font 类来美化界面 3.注意灵活使用 JPane	辅导	编程

任务 2：（20 分钟）

教学环节	教学内容	教师活动	学生活动
任务引入	点击“开始”按钮，比赛开始后枪声响起，同时运动员开始赛跑。	引入	思考
任务部署	Java 中播放音频文件 mp3 1.导入 JLayer 的 jar 包，这次我们用的是 JLayer，JLayer 是一个 Java 类库用来解码,转换,播放 MP3 文件。相对于 Java 本身提供的多媒体播放方法，这款插件功能更全面，操作更简单，我们在游戏开发的过程中，经常使用这款插件。 2.程序实现 （1）指定要播放的音频文件	讲解总结	思考

	InputStream sound=new FileInputStream(url); （2）打开一个播放器 AdvancedPlayer ap=new AdvancedPlayer(sound); （3）播放 mp3 ap.play();		

任务执行	**1.点击按钮后启动小破孩和小丫线程。**
	`public void actionPerformed(ActionEvent e) {`
	`xph.start();`
	`xy.start();`
	`}`
	点击按钮后启动线程。
		编程	思考计算

	2.在启动线程前，枪响了，然后两个运动员开始赛跑
	`public void actionPerformed(ActionEvent e) {`
	`try{`
	`InputStream sound=new FileInputStream("sound/gun.mp3");`
	`AdvancedPlayer ap=new AdvancedPlayer(sound);`
	`ap.play();`
	`}catch(Exception ex){`
	`ex.printStackTrace();`
	`}`
	枪响后，运动员开始赛跑
	思考：为什么不能用线程？
	`xph.start();`
	`xy.start();`
	赛跑开始
		演示讲解	学习修改

	3.跑步开始后，观众的加油声响了起来。思考：观众的加油声是不是应该用线程来播放？为什么？
	（1）建立线程类。思考：线程类须要定义为内部类吗？
	不需要，因为线程类不需要对主程序进行操作，它是独立的。与小破孩线程和小丫线程不同。
	`public class SoundPlayer extends Thread{`
	`String url;`
	`int n;`
	`public SoundPlayer(String url,int n){`
	`this.url=url;`
	`this.n=n;`
	`}`
	`public void run(){`
	`try{`
	`if(n>0){`
	`for(int i=0;i<n;i++){`
	`InputStream sound=new FileInputStream(url);`
	`AdvancedPlayer ap=new AdvancedPlayer(sound);`
	`ap.play();`
	`}`
	`}else{`
	`while(true){`
	`InputStream sound=new FileInputStream(url);`
	`AdvancedPlayer ap=new AdvancedPlayer(sound);`
		演示讲解	学习记录

	ap.play();
	}
	}
	}catch(Exception ex){
	ex.printStackTrace();
	}
	}
	}
	构造方法参数为 0 时，循环播放，参数>0 时，是播放次数。
	（2）生成“喝彩线程”的对象，线程对象一般定义为成员变量，这样可以在整个类的内部访问。
	Thread cheer=new SoundPlayer("sound/cheer.mp3",0);
	（3）两个运动员起跑后，开始“喝彩线程”。
	xph.start();
	xy.start();
	cheer.start();
	起跑后开始加油！		

任务 3：（10 分钟）

教学环节	教学内容	教师活动	学生活动
任务引入	比赛过程中，观众席加油，直到最后一名运动员到达终点，加油声停止。	引入	思考
任务部署	1.如何判断到达终点？ 操场的长度为 1000 像素，只要运动员的 x 值>=1000 则可以认定为到达终点。 2.如何判断两个运动员都到达终点？ 在主程序中设置计数器 n=0，用来记录，每个运动员到达终点，n++。当 n==2 时，比赛结束。	引导	思考 讨论

任务实施	1.判断运动员到达终点，则停止奔跑。
	方法是：定义标识 boolean b=true，当 x>=1000 时，b=false，退出循环。
	int x;
	boolean b=true;
	标识
	while(b){
	i++;
	x=i*20;
	if(x>=1000){
	b=false;
	}
	最后一步，要迈过终点。
		演示
		讲解
			学习
			记录

	2.每个运动员到达终点，n++，并判断 n 的值，如果 n==2，则，喝彩线程停止。
	if(x>=1000){
	b=false;
	n++;
	if(n==2){
	cheer.stop();
	}
	如果两个人都到达终点，则停止喝彩
	}
		演示
		讲解
			学习
			记录

小结	**线程停止的两种方式：** 1. cheer.stop();这种方式虽然 Java 已经淘汰，但是在某些时候还是不可取代的。比较适应于硬性终止。	分析	思考

	2.设置标识符，这种情况比较适合带循环的终止。		

任务 4：（10 分钟）

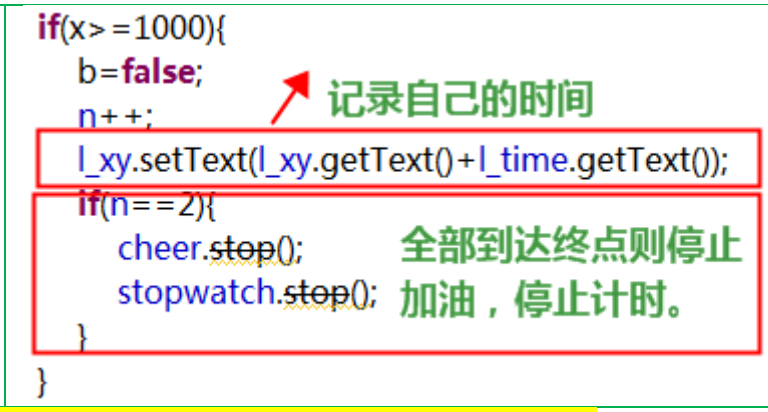
教学环节	教学内容	教师活动	学生活动

任务实施	1.定义计时器线程类
	`public class Stopwatch extends Thread{`
	`int m=0;`
	`int s=0;`
	`int mm=0;`
	`public void run(){`
	`while(true){`
	`try {`
	`Thread.sleep(100);`
	`} catch (InterruptedException e) {`
	`// TODO Auto-generated catch block`
	`e.printStackTrace();`
	`}`
	`mm++;`
	`s+=mm/10;`
	`mm=mm%10;`
	`m+=s/60;`
	`s=s%60;`
	`l_time.setText(m+":"+s+":"+mm);`
	`}`
	`}`
	`}`
	每0.1秒计时
	秒
	分
	设置显示时间
		演示讲解	学习记录

	2.声明计时器线程类对象 `Thread stopwatch=new Stopwatch();`		

	3.在枪声结束后，两个运动员开始跑步，观众开始加油，计时器开始计时，同时启用了 4 个线程。
	`xph.start();`
	`xy.start();`
	`cheer.start();`
	`stopwatch.start();`
		演示讲解	学习记录

	5.当运动员到达终点时： （1）记录自己的跑步时间，改时间为当前时刻计时器显示的时间。 （2）判断是不是最后一个到达的，如果是，则观众席停止“加油”，计时器也要停止计时。 （3）整个程序结束。	演示讲解	学习记录

Step3：总结与课后安排【5 分钟】

教学环节	教学内容	教师活动	学生活动

教学小结	**1.用 JLayer 播放声音，具体步骤为：**
	（1）指定要播放的音频文件
	（2）打开一个播放器
	（3）播放 mp3
	`try{`
	`  InputStream sound=new FileInputStream("sound/gun.mp3");`
	`  AdvancedPlayer ap=new AdvancedPlayer(sound);`
	`  ap.play();`
	`}catch(Exception ex){`
	`  ex.printStackTrace();`
	`}`
	注意：操作应该放在 try……catch 中。
	2.线程的结束有两种方式：
	（1）主动结束。设置标识符，当标识符为 false 的时候，线程自己主动结束。比较适用于有 while 循环的线程。这种方式的结束，有现成本身代码执行。
	（2）被动结束。当线程必须要结束的时候，则在线程的外部，调用 stop 方法结束线程，这种结束方式非常暴力。
		总结	听讲记录