

单元 4.1 个性化计算器

单元教学设计

学习阶段		一、实用程序开发		学时	
项目 11	个性化计算器	学时		4	
单元 11.1	个性化计算器	学时		4	
教 学 目 标	能力目标	知识目标		素质目标	
	1. 能够理解异常和处理异常 2. 能够理解计算器的原理并设计实现计算器功能	1. 理解异常的概念 2. 理解计算器的原理	课前学习	严谨务实的工作态度； 良好的心理素质； 用于克服困难的能力；	
		1. 深入理解计算器的原理和设计实现方法 2. 熟练掌握对象数组的使用 3. 熟练掌握异常的概念和错误的概念 4. 熟练掌握异常的处理方法 5. 掌握自定义异常	课上学习练习		
		1. 掌握 Java 的 Math 类的计算方法	课上学习练习		
课 前					
课 程 思 政	资料			观看纪录片《厉害了我的国》 建议：每周的班级活动中集体观看 活动：在执教云上开展专题讨论，形成云图	
	思政主题	将个人的成才梦融入实现中华民族伟大复兴的中国梦			
	思政目标	1. 引导学生感受祖国改革开放以来的飞速进步，激发对专业学习更大的动力。 2. 用中华民族历史上的道德传承，各种优秀文化思想、精神观念来拓宽学生的知识面，感受“古人的智慧”，增强国家认同感，树立行业自信。			
	主题研讨	1. 中国古代在计算工具上的领先技术和对世界科学的贡献 2. 中国近代在超级计算方面取得的成就有哪些 3. 从一无所有到世界第一，中国超级计算机是如何突破美国封锁的？			
课 前 知 识 储	目标	1. 理解异常的概念 2. 理解计算器的原理 (1) 观看视频【Scratch 玩游戏学编程】-我的大学系列-个性化计算器 (2) 用 Scratch 完成个性化计算器项目，并在网易卡搭发布 (3) 小组间互相测试“找茬”			

备	资料			
		【Scratch 玩游戏学编程】 -我的大学系列-个性化计算器-1-计算器的原理和实现	【Scratch 玩游戏学编程】 -我的大学系列-个性化计算器-2-完善计算器的功能	网易卡搭项目发布：个性化计算器源码
课上				
课 程 思 政	思政资料			
		中国古代已有“微型计算机”，钱学森靠它推算原子弹理论数据！	第五大发明：算盘也可以匹配高级计算机，现场表现开方运算	中国最早的运算工具是何，阿拉伯数字传入中国前，古人如何计数？
	主题导入 课程思政	组织学生观看关于古代计算工具的小视频，用中华民族历史上的道德传承，各种优秀文化思想、精神观念来拓宽学生的知识面，感受“古人的智慧”，增强国家认同感，树立行业自信。		
	项目导入	讨论并导入项目： 1.中国古代计算工具有那些？ 2.现在的计算工具有哪些？ 3.那些地方在使用计算器？ 4.对于复杂的运算我们怎么办？ 5.我们国家的超级计算机。		
	课前检测		1. 扫码参加异常和异常处理的 2. 根据测试结果讲解 3. 系统记录测试成绩	
能力 训练 任务	任务 1	设计计算器界面 1-画出设计图，计算坐标 2-用 Java 实现 GUI 界面设计		
	任务 2	实现计算器的基本功能		
	任务 3	用对象数组优化代码		
	任务 4	除数为 0 的异常处理		

任务 5	错误输入的异常处理		
课后			
思政资料			
	用精美图谱讲述电子计算机的发展历史	中国的智慧，西方永远不懂！20 年时间，超级计算机就做到世界第一	研究超级计算机哪家强？我国率先诞生 E 级超算，这是美国赶不上的
课程思政	用电影中、现实中的事例告诉学生，中国在信息技术行业取得了很大的进步，从追赶者变为领先者。引导学生感受祖国改革开放以来的飞速进步，激发对专业学习更大的动力。		
项目拓展	微课：Math 类的常用方法 1-连续计算功能的实现 2-幂运算、开方运算等复杂运算的功能实现		

单元教学进度设计

一、项目导入 (20) 分钟

主题导入 (5 分钟)

结合课前看的纪录片《厉害了我的国》开展主题讨论：

1. 方法：在执教云上开设研讨活动，活动主题：《厉害了我的国》中我们国家在微电子技术为基础的“计算机”、“通信”和“控制”技术这三个方面你印象最深刻的成就，活动后形成云图

2. 总结。

一是神威·太湖之光超级计算机。这台由我国并行计算机工程技术研究中心研制、安装在国家超级计算无锡中心的超级计算机，是世界上首个峰值运算速度超过十亿亿次的超级计算机，一分钟计算能力相当于全世界 72 亿人同时用计算机计算 32 年。神威·太湖之光有三项成果入围超算界的诺贝尔奖——戈登贝尔奖，并凭借其中一项最终获奖。

二是光量子计算机。2017 年 5 月 3 日，世界首台超越早期经典计算机的光量子计算机在我国诞生，为最终实现超越经典计算能力的量子计算（国际学术界称之为“量子称霸”）的目标，奠定了坚实的基础。

三是量子卫星“墨子”号。2017 年 8 月 10 日，中国科学技术大学潘建伟团队宣布，全球首颗量子科学实验卫星“墨子号”圆满完成三大科学实验任务：量子纠缠分发、量子密钥分发、量子隐形传态。目前中国量子通信技术领先国际相关技术水平 5 年，并将在未来 10 到 15 年持续保持领先。

3. 讨论最新超级计算发布

教师活动	学生活动
1. 打开职教云，展示看电影时学生门参加讨论时形成的云图	1. 回顾、讨论
2. 系统抽签，以小组为单位简述观后感	2. 推举一位同学代表小组发言

项目铺垫

从古至今我们就是一个创新的民族，在计算领域做出了很多贡献，观看小视频

		
中国古代已有“微型计算机”，钱学森靠它推算原子弹理论数据！	第五大发明：算盘也可以匹配高级计算机，现场表现开方运算	中国最早的运算工具是何，阿拉伯数字传入中国前，古人如何计数？

看视频后展开主题讨论：

1. 中国古代在计算工具上的领先技术和对世界科学的贡献
2. 中国近代在超级计算方面取得的成就有哪些
3. 从一无所有到世界第一，中国超级计算机是如何突破美国封锁的？

二、课前测试与知识梳理 (20 分钟)

课前测试		<ol style="list-style-type: none">1. 扫码参加异常和异常处理的2. 根据测试结果讲解3. 系统记录测试成绩
讲解重点	<p>Java 提供了更加优秀的解决办法：异常处理机制。</p> <p>异常处理机制能让程序在异常发生时，按照代码的预先设定的异常处理逻辑，针对性地处理异常，让程序尽最大可能恢复正常并继续执行，且保持代码的清晰。</p> <p>Java 中的异常可以是函数中的语句执行时引发的，也可以是程序员通过 <code>throw</code> 语句手动抛出的，只要在 Java 程序中产生了异常，就会用一个对应类型的异常对象来封装异常，JRE 就会试图寻找异常处理程序来处理异常。</p> <p><code>Throwable</code> 类是 Java 异常类型的顶层父类，一个对象只有是 <code>Throwable</code> 类的（直接或者间接）实例，他才是一个异常对象，才能被异常处理机制识别。JDK 中内建了一些常用的异常类，我们也可以自定义异常。</p>	

任务告知	【项目展示】
	以小组为单位展示发布在网易卡搭上的计算器，并展示除了基本功能外还实现了那些功能。
	【任务告知】
	用 Java 的 GUI 图形化编程计算实现计算器

三、任务实施

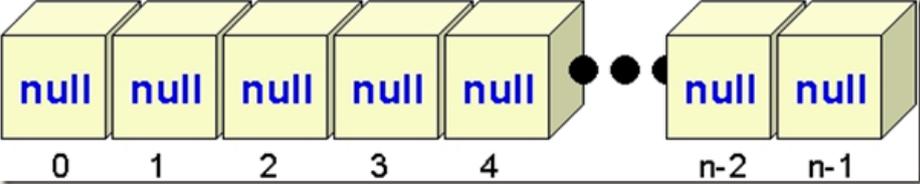
任务 1: (40 分钟)

任务部署	任务描述：设计计算器界面 1. 画出设计图，计算坐标 2. 用 Java 实现 GUI 界面设计 参考资料： 1. Java API 文档 2. 视频： 【Scratch 玩游戏学编程】-我的大学系列-个性化计算器	布置任务	参与讨论
	学生根据提示，设计满足项目需求的界面 1. 计算器界面上要有那些按钮和显示？ 2. 这些按钮和显示分别用什么组件？ 3. 画出设计图，并决定用什么布局管理器	辅导答疑	编写程序
	 第一步：画出草图  第二步：计算坐标 (3) 第三步：根据设计图和坐标写出程序。坐标的初步计算与窗口的效果会有出入，这时候就需要微调了。	点评	参与讨论
任务实施	<pre> import java.awt.Container; import javax.swing.JButton; public class ButtonFrame extends JFrame{ // 定义组件为成员变量 JButton b1=new JButton(); JButton b2=new JButton(); public void showComponents() { // 获取顶级容器上的中间容器 Container c=getContentPane(); c.setLayout(null); // 设置组件属性 b1.setBounds(20, 20, 200, 40); b1.setText("Button1"); b2.setBounds(20, 100,200, 40); b2.setText("Button2"); // 将组件放在容器上 c.add(b1); c.add(b2); } public static void main(String[] args) { ButtonFrame lf=new ButtonFrame(); lf.createFrame(); lf.showComponents(); } } </pre>		

	<p>step1: 创建窗口</p> <p>step2: 按照组件</p> <p style="padding-left: 20px;">step2.1 定义组件，一般定义为成员变量</p> <p style="padding-left: 20px;">step2.2 获取窗口的中介容器 c，并设置 c 上的布局管理器（在这里我们设置为不使用布局管理器 null）</p> <p style="padding-left: 20px;">step2.3 设置组件的属性（包括位置、大小、对齐方式等）、</p> <p style="padding-left: 20px;">step2.4 将组件安装在窗口上</p> <p>step3: 创建窗口对象，调用方法显示窗口并显示窗口上的组件</p>		
--	--	--	--

任务 3: (20 分钟)

任务引入	<p>用对象数组优化代码</p> <p>分析前面的代码，按钮有很多个，按钮的初始化、坐标的计算、事件监听器的装载等要花费大量的代码，代码重复的很高，怎样实现？</p>		
任务部署	<p>1. Java 数组的使用</p> <p>对象数组其实和 Java 的数组类似的，所以要很清楚 Java 的数组是如何使用的，如果有不懂的可以点下面这个链接 Java 数组的使用</p> <p>2. Java 的对象数组</p> <p>2.1 问题提出</p> <p>为什么会有对象数组呢？假定有这种情况，我们需要录入一个班级同学的姓名，学号，和成绩，然后对成绩进行排序，以降序或者升序进行排列，这个时候就会有人跳出来说，我用一个字符串数组存储姓名，然后再用一个二维数组存放学号和成绩，然后根据程序的高低进行排序。</p> <p>2.2 问题解析</p> <p>今天我们来教大家如何使用对象数组来解决这个问题，对象数组，我们前面学过 Java (OOP) 编程— (Java OOP 编程)，想必大家也对面向对象这个词也会稍微有了一些了解，对象数组就是可以存放多种不同的数据类型，然后通过对象来调用我们所存放的数据。</p> <p>接下来我就来告诉大家如何使用对象数组，完成这个成绩排序的问题</p> <p>2.3 问题拆分</p> <p>我们可以把问题简化一下，输入五个学生的成绩，然后进行排序打印输出</p> <p>先创建一个学生类</p> <p>给学生类添加学生的信息—姓名，学号，成绩，然后分别生成有参和无参的构造方法</p> <p>再创建一个学生测试类</p> <p>创建对象数组，给对象数组申请 5 个空间</p> <p>循环录入信息</p> <p>根据学生成绩进行排序，然后打印输出</p>	演示讲解	听讲模仿思考

	<p>请录入学生姓名，学号、成绩：</p> <pre>Alice 001 96 Bob 002 95 Gorit 003 100 Mary 004 99 Li 005 94</pre> <p>学生的成绩排名如下</p> <table border="1"> <thead> <tr> <th>姓名</th> <th>学号</th> <th>成绩</th> </tr> </thead> <tbody> <tr> <td>Gorit</td> <td>3</td> <td>100</td> </tr> <tr> <td>Mary</td> <td>4</td> <td>99</td> </tr> <tr> <td>Alice</td> <td>1</td> <td>96</td> </tr> <tr> <td>Bob</td> <td>2</td> <td>95</td> </tr> <tr> <td>Li</td> <td>5</td> <td>94</td> </tr> </tbody> </table> 	姓名	学号	成绩	Gorit	3	100	Mary	4	99	Alice	1	96	Bob	2	95	Li	5	94		
姓名	学号	成绩																			
Gorit	3	100																			
Mary	4	99																			
Alice	1	96																			
Bob	2	95																			
Li	5	94																			
<p>任务实施</p>	<p>代码【略】</p> <p>注意：</p> <ol style="list-style-type: none"> 1. 在定义对象数组时，思考在什么时候开辟的空间？ 2. 对象数组的空间分布情况？ 3. 每个 JButton 在数组中存放的时地址还是对象本身？ 4. 注意定义方法让程序更有结构感。 	<p>辅导</p>	<p>编程</p>																		

任务 4: (40 分钟)

<p>任务引入</p>	<p>除数为 0 的异常处理</p> <p>错误的情境：进行除法运算时，除数输入 0，结果程序报错，无法继续运行。</p> <p>功能比较：打开计算机上的计算器，除数输入 0 后，计算器报错</p>	<p>引入</p>	<p>思考</p>
	<p>讨论：怎样处理这个问题</p> <p>解决方案 1：做除法运算前判断除数是否为 0，如果除数是 0，则报错</p> <p>Scratch 模拟这部分功能</p> <p>解决方案 2：异常处理</p>		
<p>课程</p>	<p>在异常处理中融入民族自豪感、爱国心</p>		

思政

因为在以前的课程中，没有涉及异常的处理，所以学生在理解上会有些困难。讲授相关知识点时，可以在设计案例中采用类比的方式来进行。比如，5月21日夜間、22日凌晨，云南漾濞、青海玛多相继发生强震大震，突如其来的地震这一自然灾害就像程序中的异常，我们的党和国家是如何 catch 这个“异常”并对这个“异常”进行处理的呢？

地震发生后，国家及时 catch 了这个“异常”并对“异常”采取了有力的措施，迅速组织各种力量进行救灾，人民子弟兵、消防战士、武警战士、医护人员、通信人员还有全国各地的志愿者们纷纷赶往灾区，去援救被围困的人们，全力以赴地抢救伤员，最大限度地减少人员伤亡，保护了人民生命财产的安全。“finally”交出了一份让全国人们满意的答卷，到目前为止，灾后重建工作也基本完成。这充分体现了我们国家体制的优越性，使我们深深感受到身作为一名中国人是多么的幸福。作为一个有着 14 亿人口的大国，我们伟大的祖国从来未放弃过任何一个国人，积极地采取各种应对措施，人民的教育、生活、医疗一项也未曾落下。哪有什么岁月静好，只不过是祖国在替我们负重前行。

通过我们对地震这个自然灾害的应急处理，让同学们感受到国家强大的力量，感受到身为中国人的自豪和骄傲，从而激发自身强大的爱国热情。并开展讨论：

1. 用异常处理的思路，举例说明我们国家还应急处理过那些重大事故？
2. 在事故处理的过程中，那些是 try，那些是 catch，那些是 finally
3. 处理事故的时候，finally 是否是必须处理的？

题	题	题
四川遭遇强降雨 消防开展 抗洪救灾	河南卫辉农村救灾：顿坊 店乡 3000 多人被转移安置	伟大的抗震救灾精神：一方 有难，八方支援！

观看视频，用异常处理的“语法”描述这几个事件。

任务部署

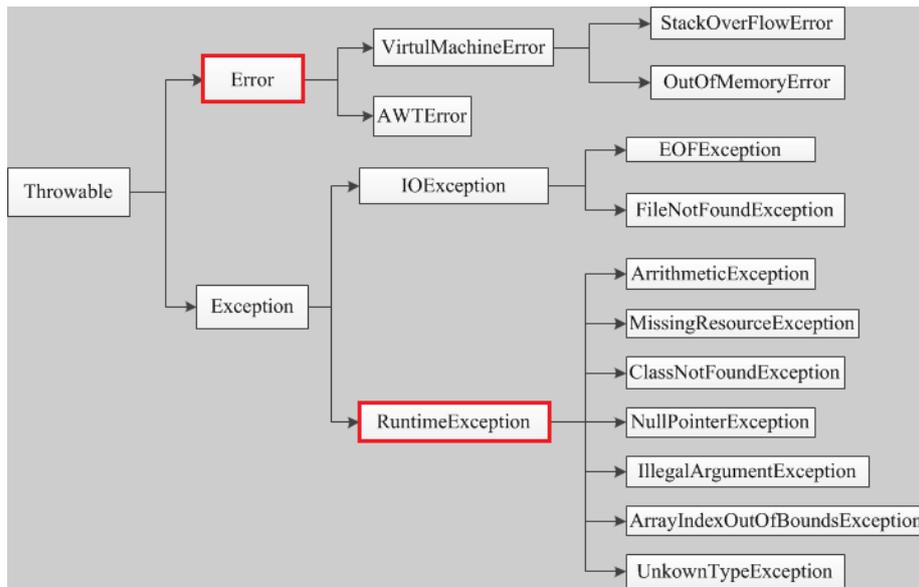
Java 标准库内建了一些通用的异常，这些类以 Throwable 为顶层父类。

Throwable 又派生出 Error 类和 Exception 类。

错误：Error 类以及他的子类的实例，代表了 JVM 本身的错误。错误不能被程序员通过代码处理，Error 很少出现。因此，程序员应该关注 Exception 为父类的分支下的各种异常类。

异常：Exception 以及他的子类，代表程序运行时发送的各种不希望发生的事件。可以被 Java 异常处理机制使用，是异常处理的核心。

讲解 学习



总体上我们根据 Javac 对异常的处理要求，将异常类分为 2 类。

非检查异常 (unchecked exception): Error 和 RuntimeException 以及他们的子类。

javac 在编译时，不会提示和发现这样的异常，不要求在程序处理这些异常。所以如果愿意，我们可以编写代码处理（使用 try... catch... finally）这样的异常，也可以不处理。对于这些异常，我们应该修正代码，而不是去通过异常处理器处理。这样的异常发生的原因多半是代码写的有问题。如除 0 错误 ArithmeticException，错误的强制类型转换错误 ClassCastException，数组索引越界 ArrayIndexOutOfBoundsException，使用了空对象 NullPointerException 等等。

检查异常 (checked exception): 除了 Error 和 RuntimeException 的其它异常。javac 强制要求程序员为这样的异常做预备处理工作（使用 try... catch... finally 或者 throws）。在方法中要么用 try-catch 语句捕获它并处理，要么用 throws 子句声明抛出它，否则编译不会通过。这样的异常一般是由程序的运行环境导致的。因为程序可能被运行在各种未知的环境下，而程序员无法干预用户如何使用他编写的程序，于是程序员就应该为这样的异常时刻准备着。如 SQLException，IOException, ClassNotFoundException 等。

需要明确的是：检查和非检查是针对 javac 来说的，这样就很好理解和区分了。

异常是在执行某个函数时引发的，而函数又是层级调用，形成调用栈的，因为，只要一个函数发生了异常，那么他的所有的 caller 都会被异常影响。当这些被影响的函数以异常信息输出时，就形成了 **异常追踪栈**。

异常最先发生的地方，叫做 **异常抛出点**。



从上面的例子可以看出，当 `divide` 函数发生除 0 异常时，`divide` 函数将抛出 `ArithmeticException` 异常，因此调用他的 `CMDCalculate` 函数也无法正常完成，因此也发送异常，而 `CMDCalculate` 的 caller——`main` 因为 `CMDCalculate` 抛出异常，也发生了异常，这样一直向调用栈的栈底回溯。这种行为叫做异常的冒泡，异常的冒泡是为了在当前发生异常的函数或者这个函数的 caller 中找到最近的异常处理程序。由于这个例子中没有使用任何异常处理机制，因此异常最终由 `main` 函数抛给 JRE，导致程序终止。

上面的代码不使用异常处理机制，也可以顺利编译，因为 2 个异常都是非检查异常。但是下面的例子就必须使用异常处理机制，因为异常是检查异常。

代码中我选择使用 `throws` 声明异常，让函数的调用者去处理可能发生的异常。但是为什么只 `throws` 了 `IOException` 呢？因为 `FileNotFoundException` 是 `IOException` 的子类，在处理范围内。

```
public void testException() throws IOException
{
    //FileInputStream 的构造函数会抛出 FileNotFoundException
    FileInputStream fileIn = new FileInputStream("E:\\a.txt");

    int word;
    //read 方法会抛出 IOException
    while((word = fileIn.read())!=-1)
    {
        System.out.print((char)word);
    }
    //close 方法会抛出 IOException
    fileIn.close()
}
```

异常处理的基本语法

在编写代码处理异常时，对于检查异常，有 2 种不同的处理方式：使用 `try...catch...finally` 语句块处理它。或者，在函数签名中使用 `throws` 声明交给函数调用者 `caller` 去解决。

try...catch...finally 语句块

```
try{
    //try 块中放可能发生异常的代码。
    //如果执行完 try 且不发生异常，则接着去执行 finally 块和 finally 后面的代码（如果有的话）。
    //如果发生异常，则尝试去匹配 catch 块。
}catch(SQLException SQLException){
    //每一个 catch 块用于捕获并处理一个特定的异常，或者这异常类型的子类。Java7 中可以将多个异常声明在一个 catch 中。
    //catch 后面的括号定义了异常类型和异常参数。如果异常与之匹配且是最先匹配到的，则虚拟机将使用这个 catch 块来处理异常。
```

//在 catch 块中可以使用这个块的异常参数来获取异常的相关信息。异常参数是这个 catch 块中的局部变量，其它块不能访问。

//如果当前 try 块中发生的异常在后续的所有 catch 中都没捕获到，则先去执行 finally，然后到这个函数的外部 caller 中去匹配异常处理器。

//如果 try 中没有发生异常，则所有的 catch 块将被忽略。

```
}catch(Exception exception){
```

```
    //...
```

```
}finally{
```

```
    //finally 块通常是可选的。
```

```
    //无论异常是否发生，异常是否匹配被处理，finally 都会执行。
```

```
    //一个 try 至少要有 1 个 catch 块，否则，至少要有 1 个 finally 块。但是 finally 不是用来处理异常的，finally 不会捕获异常。
```

```
    //finally 主要做一些清理工作，如流的关闭，数据库连接的关闭等。
```

```
}
```

需要注意的地方

1、try 块中的局部变量和 catch 块中的局部变量（包括异常变量），以及 finally 中的局部变量，他们之间不可共享使用。

2、每一个 catch 块用于处理一个异常。异常匹配是按照 catch 块的顺序从上往下寻找的，只有第一个匹配的 catch 会得到执行。匹配时，不仅运行精确匹配，也支持父类匹配，因此，如果同一个 try 块下的多个 catch 异常类型有父子关系，应该将子类异常放在前面，父类异常放在后面，这样保证每个 catch 块都有存在的意义。

3、java 中，异常处理的任务就是将执行控制流从异常发生的地方转移到能够处理这种异常的地方去。也就是说：当一个函数的某条语句发生异常时，这条语句的后面的语句不会再执行，它失去了焦点。执行流跳转到最近的匹配的异常处理 catch 代码块去执行，异常被处理完后，执行流会接着在“处理了这个异常的 catch 代码块”后面接着执行。

有的编程语言当异常被处理后，控制流会恢复到异常抛出点接着执行，这种策略叫做：

resumption model of exception handling（恢复式异常处理模式）

而 Java 则是让执行流恢复到处理了异常的 catch 块后接着执行，这种策略叫做：

termination model of exception handling（终结式异常处理模式）

```
public static void main(String[] args){
    try {
        foo();
    }catch(ArithmeticException ae) {
        System.out.println("处理异常");
    }
}
public static void foo(){
    int a = 5/0; //异常抛出点
    System.out.println("为什么还不给我涨工资!!!"); ////////////////不
    会执行
}
```

<p>任务部署 3</p>	<h2 style="background-color: #333; color: white; padding: 5px;">finally 块</h2> <p>inally 块不管异常是否发生，只要对应的 try 执行了，则它一定也执行。只有一种方法让 finally 块不执行：<code>System.exit()</code>。因此 finally 块通常用来做资源释放操作：关闭文件，关闭数据库连接等等。</p> <p>良好的编程习惯是：在 try 块中打开资源，在 finally 块中清理释放这些资源。</p> <p>需要注意的地方：</p> <ol style="list-style-type: none"> 1、finally 块没有处理异常的能力。处理异常的只能是 catch 块。 2、在同一 try...catch...finally 块中，如果 try 中抛出异常，且有匹配的 catch 块，则先执行 catch 块，再执行 finally 块。如果没有 catch 块匹配，则先执行 finally，然后去外面的调用者中寻找合适的 catch 块。 3、在同一 try...catch...finally 块中，try 发生异常，且匹配的 catch 块中处理异常时也抛出异常，那么后面的 finally 也会执行：首先执行 finally 块，然后去外围调用者中寻找合适的 catch 块。 <p>这是正常的情况，但是也有特例。关于 finally 有很多恶心，偏、怪、难的问题，我在本文最后统一介绍了，电梯速达->: finally 块和 return</p>	<p>布置任务</p>	<p>思考</p>
<p>任务部署 4</p>	<h2 style="background-color: #333; color: white; padding: 5px;">throw 异常抛出语句</h2> <p>throw <code>exceptionObject</code></p> <p>程序员也可以通过 throw 语句手动显式的抛出一个异常。throw 语句的后面必须是一个异常对象。</p> <p>throw 语句必须写在函数中，执行 throw 语句的地方就是一个异常抛出点，它和由 JRE 自动形成的异常抛出点没有任何差别。</p> <pre>public void save(User user) { if(user == null) throw new IllegalArgumentException("User 对象为空"); //..... }</pre>		
<p>任务实施</p>	<p>代码【略】</p> <p>注意：</p> <ol style="list-style-type: none"> 1. 程序的结构要清晰 2. 定义方法去调用，让程序结构感更强。 	<p>点评</p>	<p>展示</p>

任务 5: (20 分钟)

任务引入	<p>错误输入的异常处理</p> <p>错误情境设置:</p> <p>输入乘法运算时, 没有输入第一操作数, 直接输入运算符, 则运算错误。</p> <p>输入减法运算时, 没有输入操作数和运算符, 直接点等号, 则运算错误</p> <p>讨论: 怎样解决这些错误输入的问题?</p>	展示	思考
任务部署	<p>引导学生打开电脑或手机上的计算器, 讨论各种错误的解决方案</p> <ol style="list-style-type: none"> 1.以小组为单位互相“抓虫子” 2.记录存在的问题, 讨论解决办法 	布置任务	小组讨论
任务实施	<p>小组为单位, 进行各种正确的、错误的输入, 看计算器做了什么样的处理。</p> <p>思考:</p> <ol style="list-style-type: none"> 1. 那些问题是需要做错误处理的, 那些不需要 2. 那些错误是要做异常处理的, 那些不需要 <p>讨论和展示:</p> <p>以小组为单位, 展示讨论的结果, 计入小组过程考核</p>	布置任务	小组讨论

四、总结与课后安排

<ol style="list-style-type: none"> 1. 异常和异常处理的概念 2. 自定义异常处理 3. 计算器的原理 4. 对象数组 	总结	听讲记录
<p>课后拓展</p> <p>微课: Math 类的常用方法</p> <ol style="list-style-type: none"> 1-连续计算功能的实现 2-幂运算、开方运算等复杂运算的功能实现 	平台发布	小组互评