

第三讲 Java异常处理

初版时间: 2017年5月1日

初版制作者: 林焕彬

教材版本号: szlanyou-V1.1

修订时间:

修订者:

教材版本号: szlanyou-V1.1

目录



异常的概念

异常处理

声明/抛出异常

自定义异常

总结

一、生活中的异常 1-1

- 正常情况下，小王每日开车去上班，耗时大约30分钟



一路畅通



- 但是，异常情况迟早要发生！



堵车！

撞车！

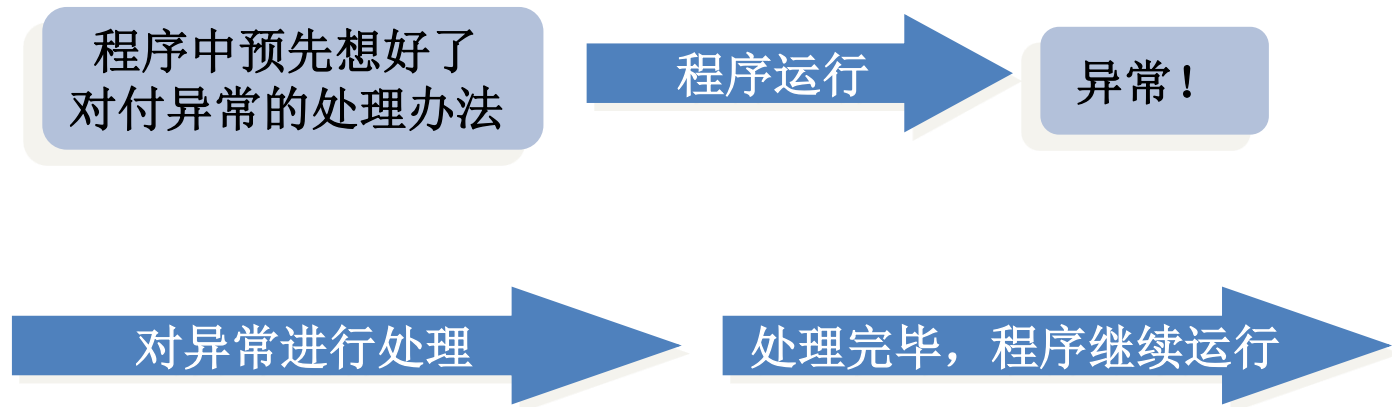


一、异常的概念 1-2

异常

- 在程序运行过程中可能发生的不正常的事件
- 会中断正在运行的程序

Java使用异常处理机制为程序提供了错误处理的能力



一、程序中的异常 1-3

```
public class Test1 {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        System.out.print("请输入被除数:");  
        int num1 = in.nextInt();  
        System.out.print("请输入除数:");  
        int num2 = in.nextInt();  
        System.out.println(String.format("%d / %d = %d",  
                                           num1, num2, num1 / num2));  
        System.out.println("感谢使用本程序！");  
    }  
}
```

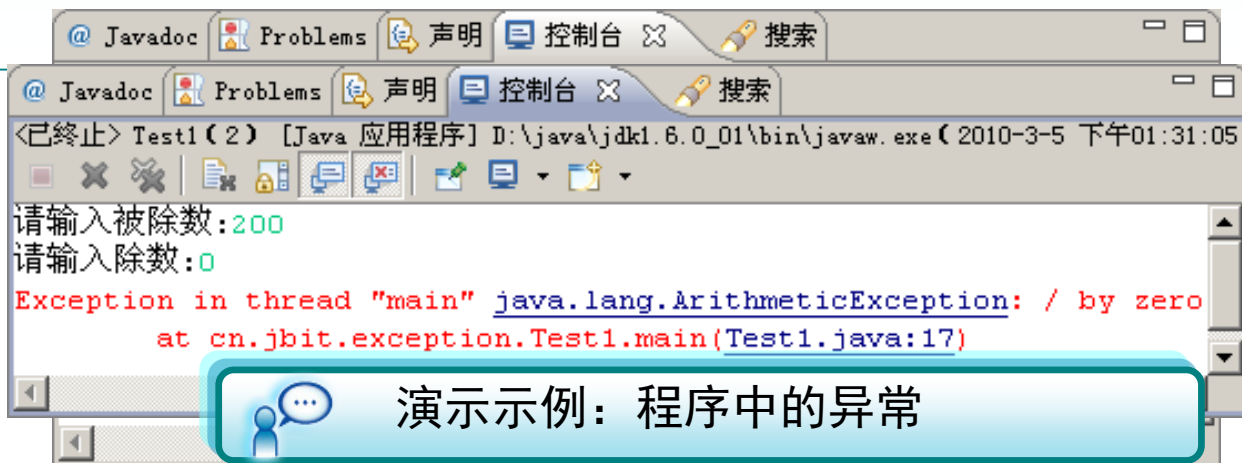
正常情况:

输入: 200 40

异常情况:

输入: B

输入: 200 0



目录



异常的概念

异常处理

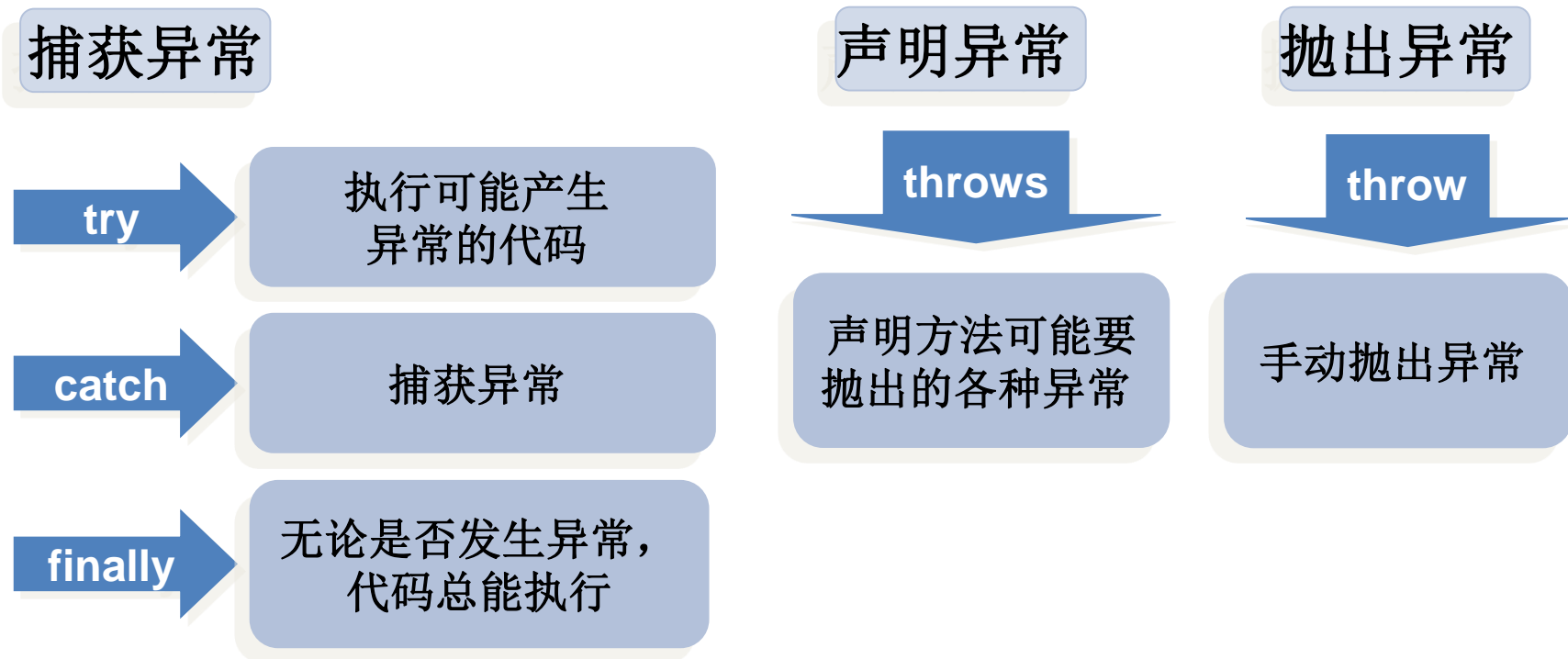
声明/抛出异常

自定义异常

总结

二、五个关键字 2-1

- Java的异常处理是通过5个关键字来实现的：**try**、**catch**、**finally**、**throw**、**throws**



二、异常处理 2-2

- 使用try-catch块捕获异常，分为三种情况：

第一种情况

```
public void method(){  
→ try {  
    // 代码段(此处不会产生异常)  
} catch (异常类型 ex) {  
    // 对异常进行处理的代码段  
}  
→ // 代码段  
}
```

try

catch

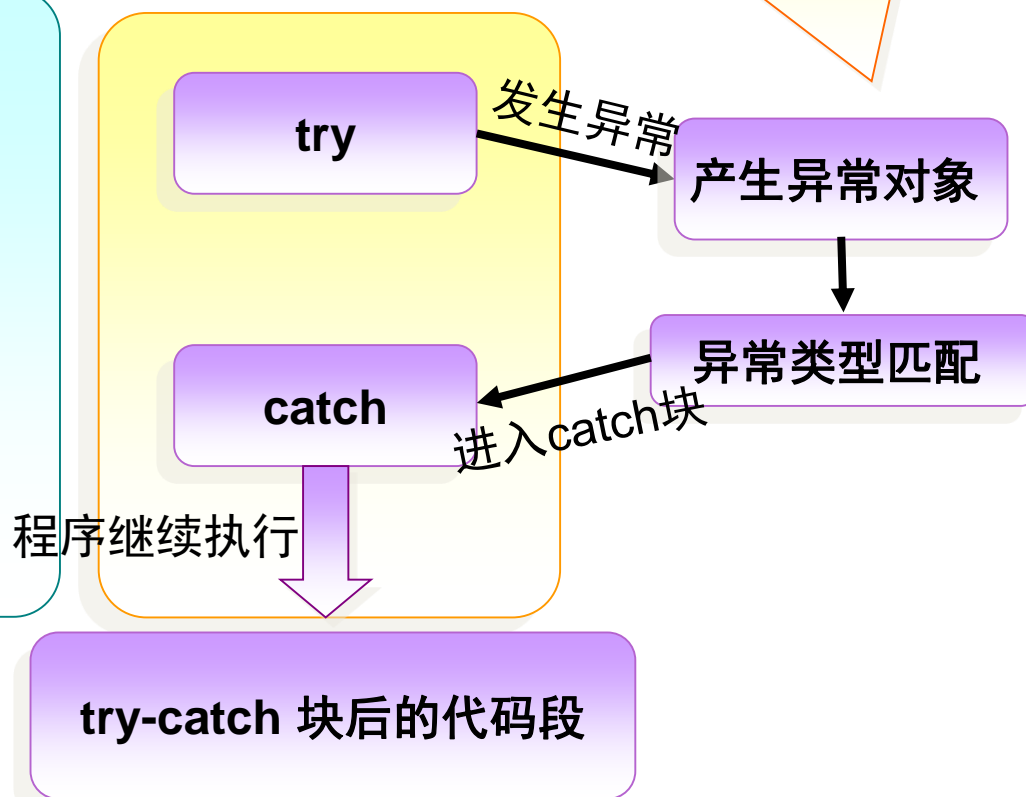
try-catch 块后的代码段

二、异常处理 2-3

第二种情况

```
public void method(){  
→ try {  
    // 代码段 1  
→    // 产生异常的代码段 2  
    // 代码段 3  
→ } catch (异常类型 ex) {  
→    // 对异常进行处理的代码段4  
    }  
→ // 代码段5  
}
```

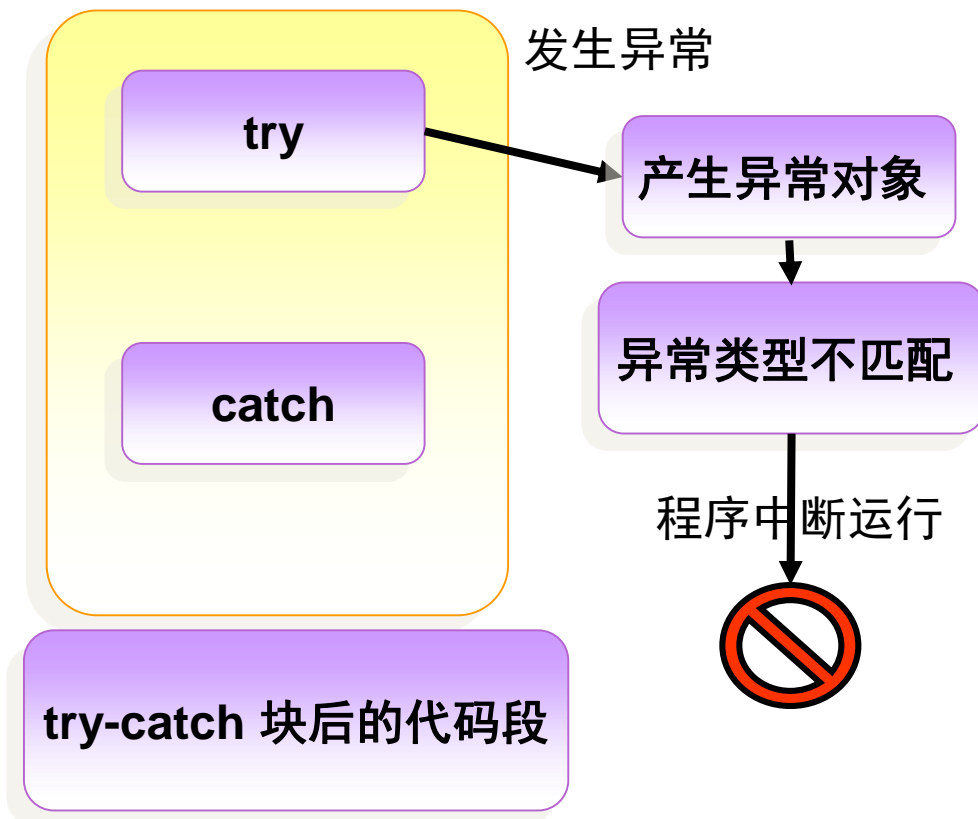
异常是一种特殊的对象，类型为 `java.lang.Exception` 或其子类



二、异常处理 2-4

第三种情况

```
public void method(){  
    try {  
        // 代码段 1  
        // 产生异常的代码段 2  
        // 代码段 3  
    } catch (异常类型 ex) {  
        // 对异常进行处理的代码段4  
    }  
    // 代码段5  
}
```

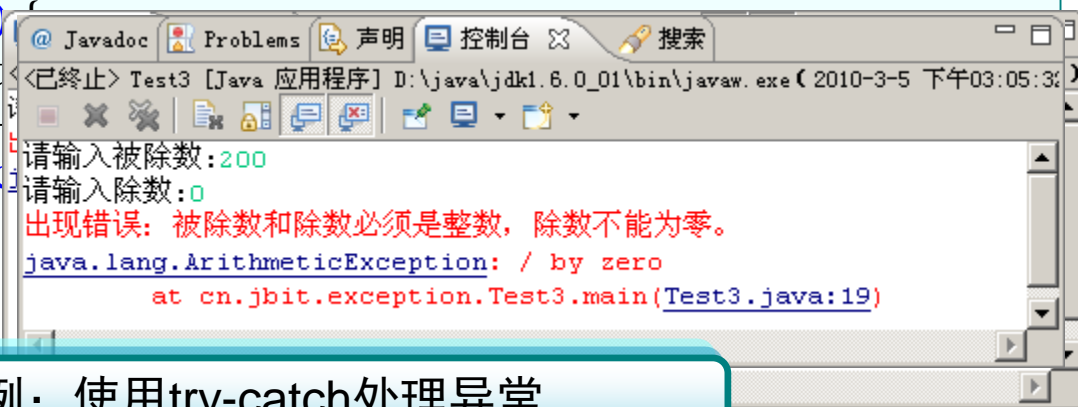


二、异常处理 2-5

```
public class Test3 {  
    public static void main(String[] args) {  
        try {  
            Scanner in = new Scanner(System.in);  
            System.out.print("请输入被除数:");  
            int num1 = in.nextInt();  
            System.out.print("请输入除数:");  
            int num2 = in.nextInt();  
            System.out.println(String.format("%d / %d = %d",  
                num1, num2, num1 / num2));  
            System.out.println("感谢使用本程序! ");  
        } catch (Exception e) {  
            System.err.println(e.getMessage());  
            e.printStackTrace();  
        }  
    }  
}
```

输入: B

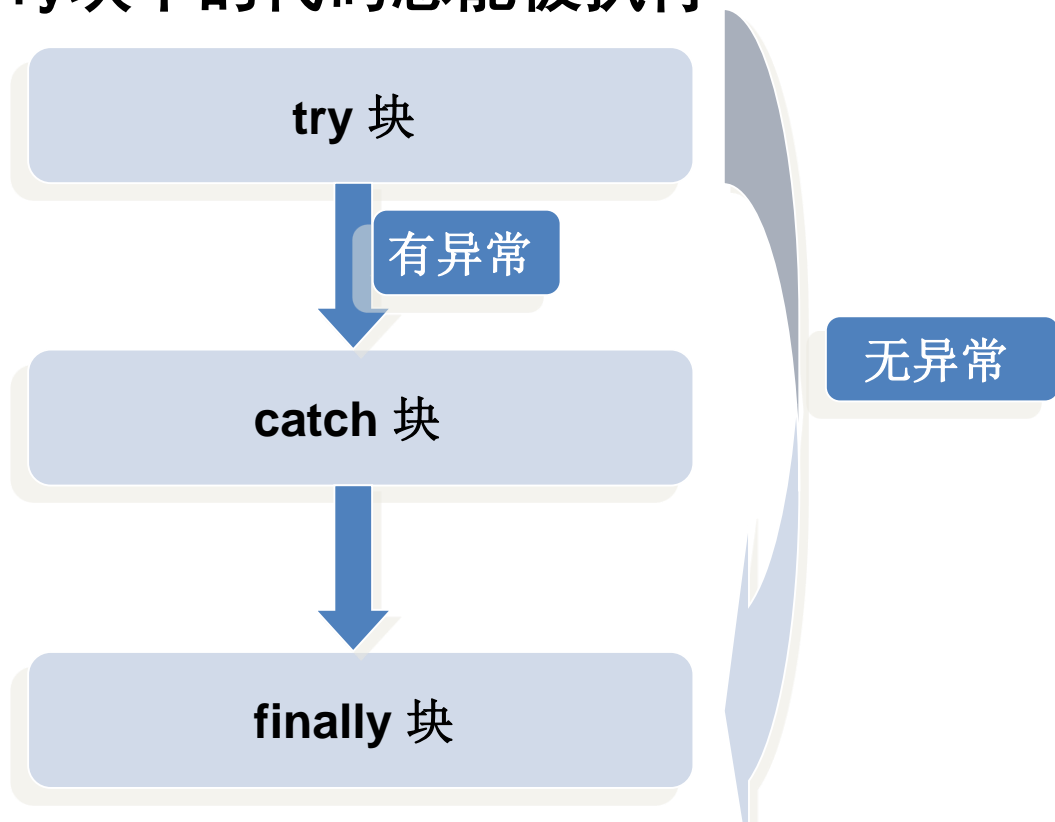
输入: 200,0



演示示例: 使用try-catch处理异常

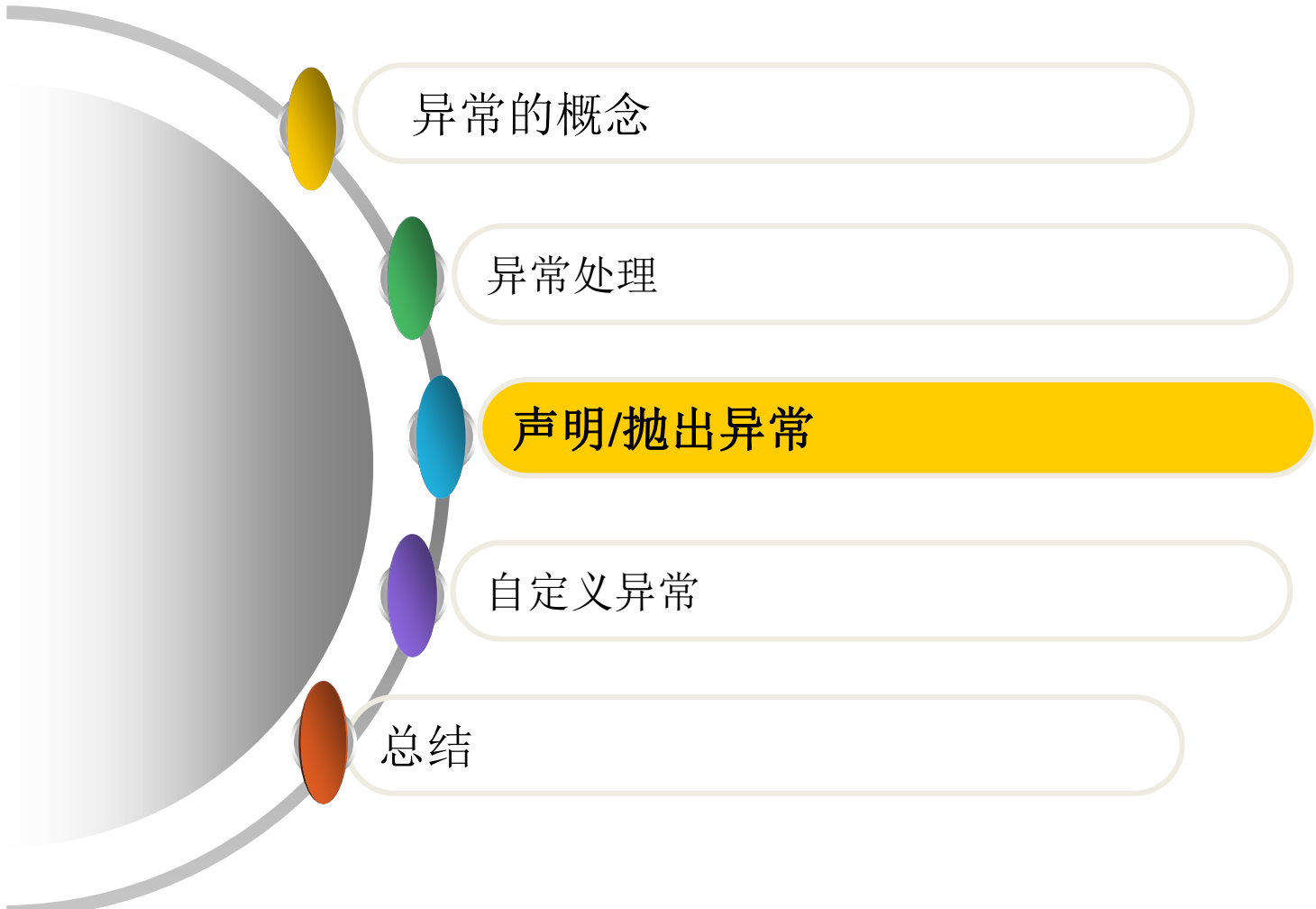
二、异常处理 2-6

- 在try-catch块后加入finally块，可以确保无论是否发生异常，finally块中的代码总能被执行



演示示例：使用try-catch-finally处理异常

目录



异常的概念

异常处理

声明/抛出异常

自定义异常

总结

三、throws声明异常 3-1

- Java语言中通过throws声明某个方法可能抛出的各种异常，可以同时声明多个异常，由逗号隔开

```
public static void divide() throws Exception {  
    //.....  
}
```

divide()方法没有处理异常，而是声明异常

```
public static void main(String[] args) {  
    try {  
        divide();  
    } catch (Exception e) { //..... }  
}
```

方式1：调用者通过try-catch捕获并处理异常

```
// public static void main(String[] args) throws Exception {  
//     divide();  
// }
```

方式2：调用者通过throws继续声明异常



演示示例：使用throws声明异常

三、throw抛出异常 3-2

- 除了系统自动抛出异常外，有些问题需要程序员自行抛出异常

```
public class Person {  
    private String sex = "男";// 性别  
    public void setSex(String sex) throws Exception {  
        if ("男".equals(sex) || "女".equals(sex))  
            this.sex = sex;  
        else {  
            throw new Exception("性别必须是“男”或者“女”！");  
        }  
    }  
}
```

抛出异常

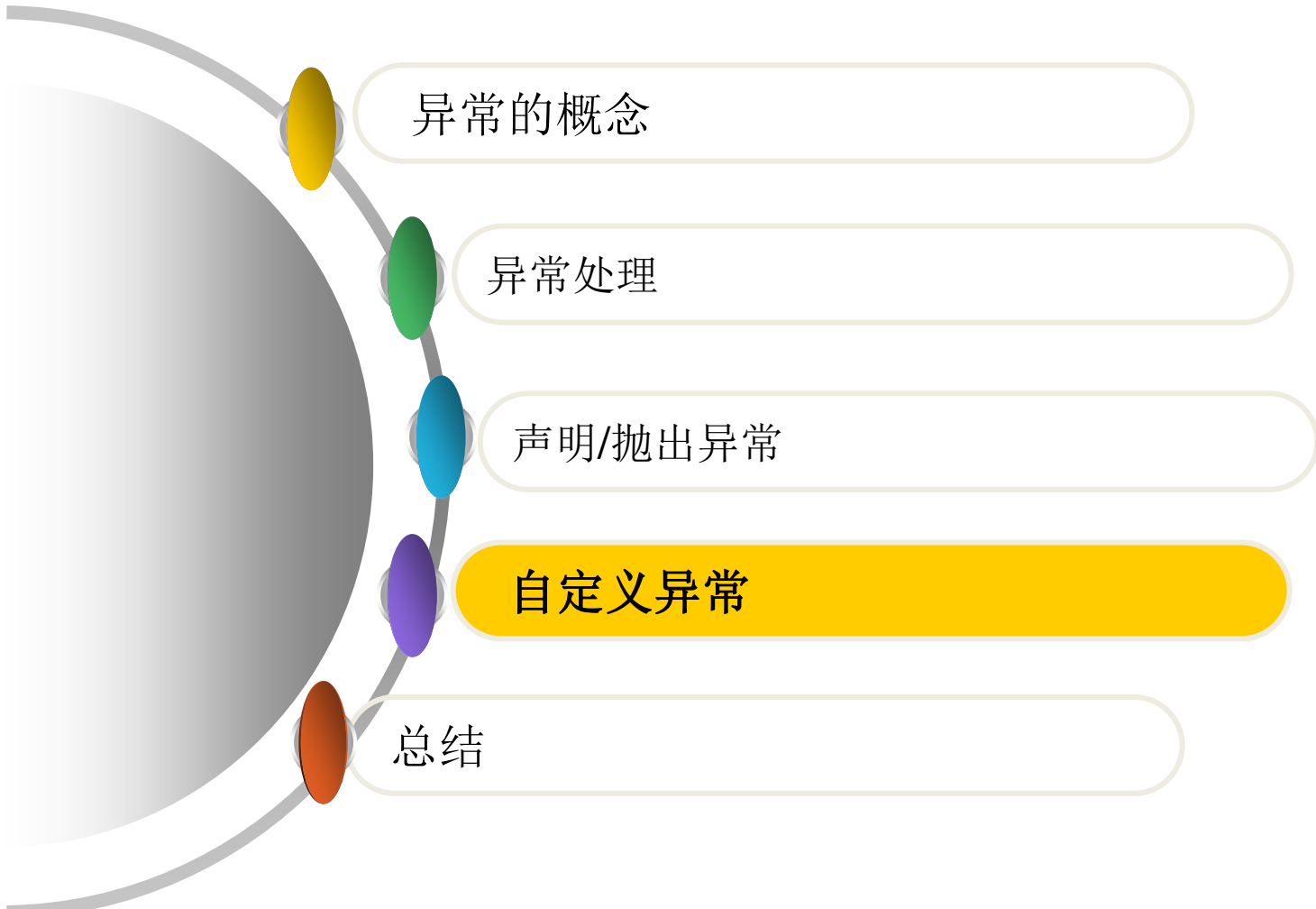
```
Person person = new Person();  
try {  
    person.setSex("Male");  
    person.print();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

捕获异常，或者
throws异常



演示示例：使用throw抛出异常

目录



异常的概念

异常处理

声明/抛出异常

自定义异常

总结

四、自定义异常 4-1



何时需要自定义异常？

- 当JDK 中的异常类型不能满足程序的需要时，可以自定义异常类
- 使用自定义异常的步骤：



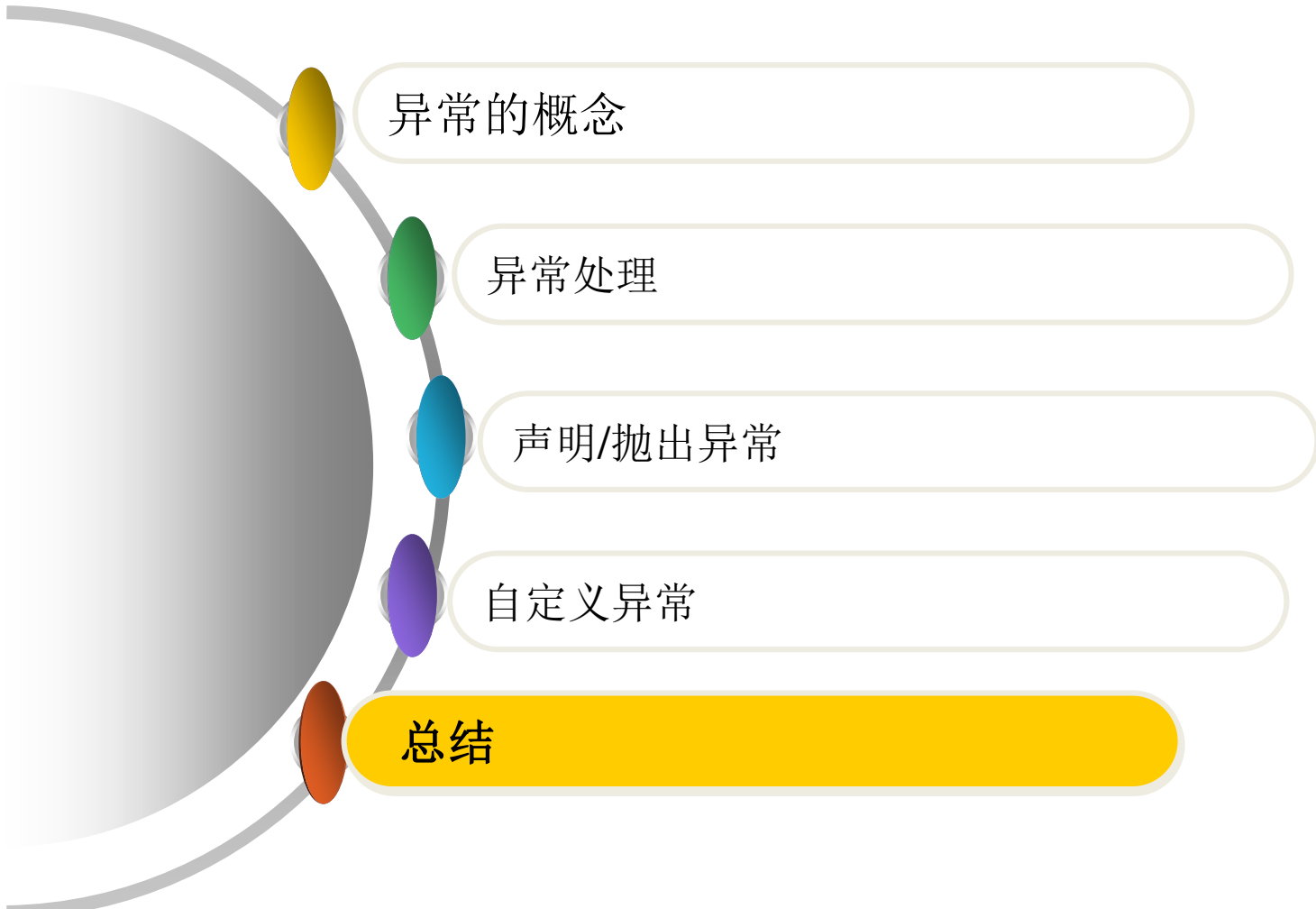
继承Exception 或者RuntimeException

 演示示例：自定义异常

四、自定义异常 4-2

```
public class BusinessException extends RuntimeException {  
  
    public BusinessException() {  
        super();  
    }  
    public BusinessException(String message) {  
        super(message);  
    }  
    public BusinessException(Throwable cause) {  
        super(cause);  
    }  
    public BusinessException(String message, Throwable  
cause) {  
        super(message, cause);  
    }  
}
```

目录



异常的概念

异常处理

声明/抛出异常

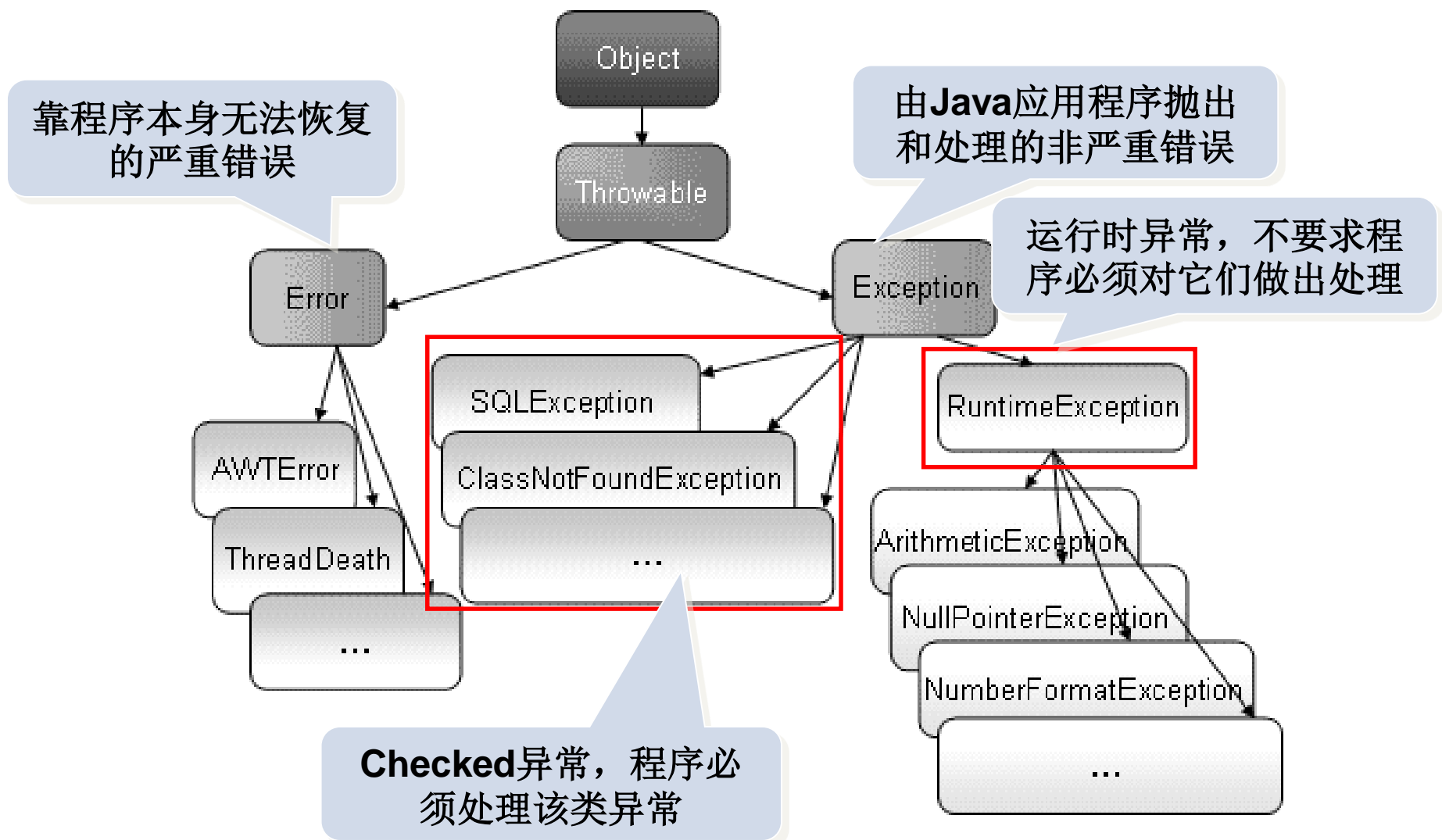
自定义异常

总结

五、常见异常总结 5-1

方 法 名	说 明
Exception	异常层次结构的根类
ArithmeticException	算术错误情形，如以零作除数
ArrayIndexOutOfBoundsException	数组下标越界
NullPointerException	尝试访问 null 对象成员
ClassNotFoundException	不能加载所需的类
InputMismatchException	欲得到数据类型与实际输入类型不匹配
IllegalArgumentException	方法接收到非法参数
ClassCastException	对象强制类型转换出错
NumberFormatException	数字格式转换异常，如把“abc”转换成数字

五、异常分类总结 5-2



五、总结



提问

- 1、异常是在程序运行过程中可能发生的不正常的事件，异常不是错误。
- 2、try-catch的三种情况。
- 3、简述throws和throw的区别。
- 4、简述自定义异常的步骤。

五、上机练习



需求说明

- 自定义异常。
- 在`setAge(int age)`中对年龄进行判断，如果年龄介于1到100直接赋值，否则抛出自定义异常。
- 在测试类中创建对象并调用`setAge(int age)`方法，使用try-catch捕获并处理异常

完成时间：10分钟

共性问题集中讲解

Thank you

Q&A

