



黑马程序员™  
[www.itheima.com](http://www.itheima.com)

传智播客旗下  
高端IT教育品牌



# 异常

# 目录 Contents

- ◆ 异常概述
- ◆ 异常处理方式
- ◆ 自定义异常

思考：

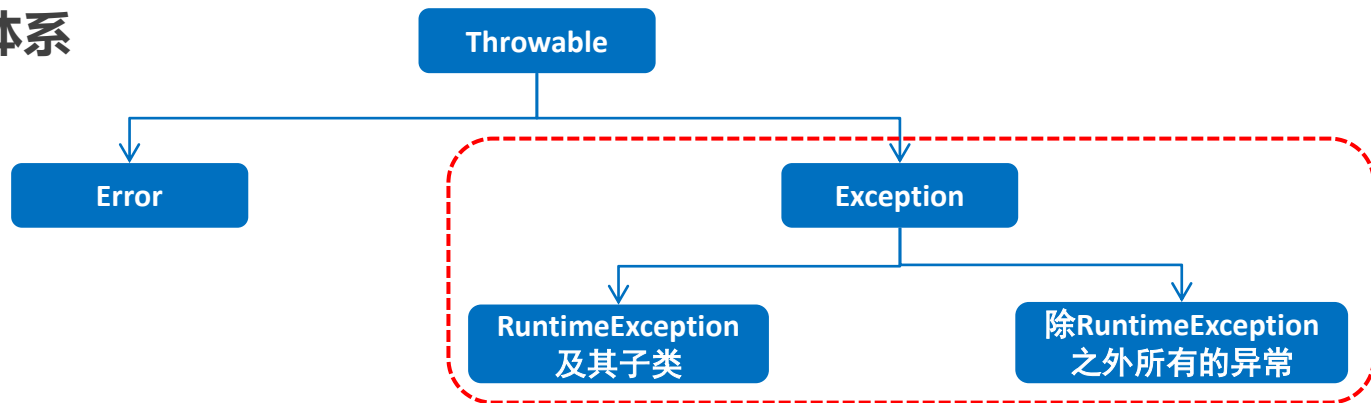
当代码出现了问题，程序是如何体现的？

**异常**就是用来描述代码中出现的问题

异常：就是程序出现了不正常的情况。程序在执行过程中，出现的非正常的情况，最终会导致 JVM 的非正常停止。

**注意：**语法错误不算在异常体系中。

## 异常体系



**Error：**严重问题，通过代码无法处理。  
比如：内存溢出。

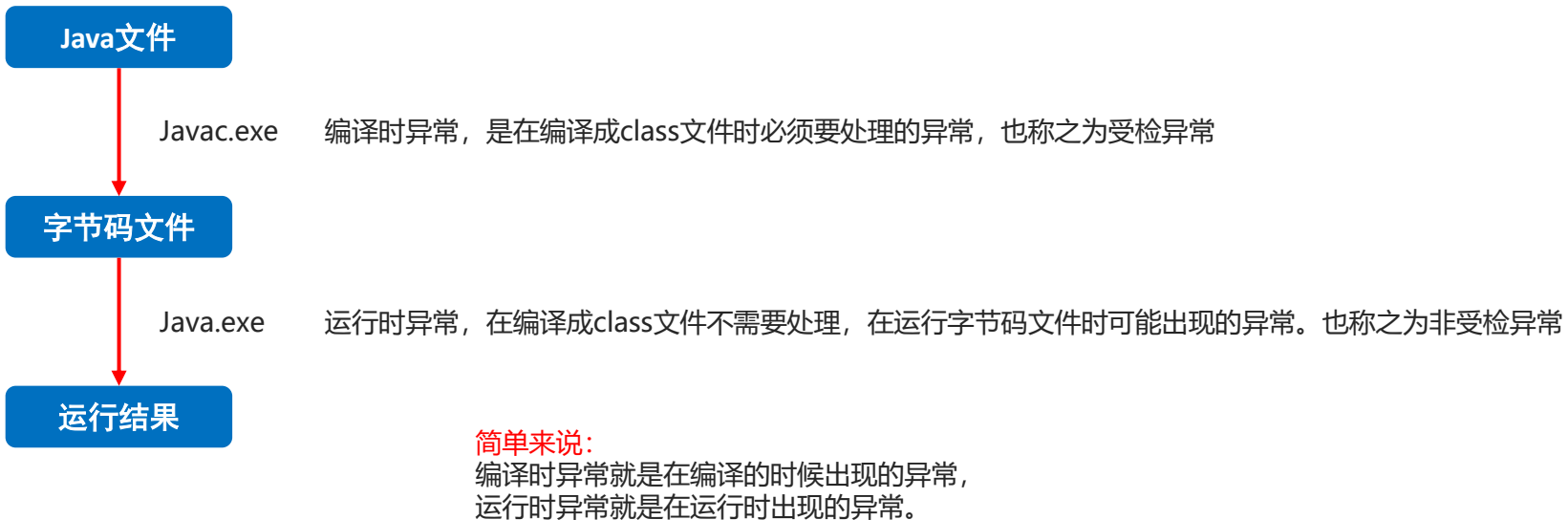
**Exception：**称为异常类，它表示程序本身可以处理的问题

- **RuntimeException及其子类：**运行时异常。（空指针异常，数组索引越界异常）
- **除RuntimeException之外所有的异常：**编译期必须处理的，否则程序不能通过编译。（日期格式化异常）。

异常：就是程序出现了不正常的情况。程序在执行过程中，出现的非正常的情况，最终会导致 JVM 的非正常停止。

**注意：**语法错误不算在异常体系中。

## 编译时异常和运行时异常



异常：就是程序出现了不正常的情况。程序在执行过程中，出现的非正常的情况，最终会导致 JVM 的非正常停止。

**注意：**语法错误不算在异常体系中。

## JVM 的默认处理方案

如果程序出现了问题，我们没有做任何处理，最终 JVM 会做默认的处理。

- 把异常的名称，异常原因及异常出现的位置等信息输出在了控制台
- 程序停止执行

异常：就是程序出现了不正常的情况。程序在执行过程中，出现的非正常的情况，最终会导致 JVM 的非正常停止。

**注意：**语法错误不算在异常体系中。

## JVM 的默认处理方案

阅读异常信息的练习

- 阅读单个方法中出现的异常
- 阅读多个方法中出现的异常
- 阅读多个类多个方法中出现的异常

## 小结

1, 异常就是程序出现的不正常的情况。

2, 异常的两类?

编译时异常、运行时异常

3, 虚拟机默认处理异常的方式?

将异常信息以红色字体打印在控制台、结束程序运行



# 目录 Contents

- ◆ 异常概述
- ◆ 异常处理方式
- ◆ 自定义异常

## 异常处理方式 —— throws

格式:

```
throws 异常类名;
```

**注意:** 这个格式是写在方法的定义处，表示声明一个异常。

- 编译时异常因为在编译时就会检查，所以必须要写在方法后面进行显示声明
- 运行时异常因为在运行时才会发生，所以在方法后面可以不写

## 抛出异常对象 —— throw

思考：

以前出现了异常，虚拟机帮我们创建一个异常对象，抛给调用者。但是如果我们需要自己手动创建一个异常对象该如何写？

格式：

```
throw new 异常();
```

**注意：**这个格式是在方法内的，表示当前代码手动抛出一个异常，下面的代码不用再执行了。

## throws 和 throw 的区别

### throws

- 用在方法声明后面，跟的是异常类名
- 表示声明异常，调用该方法有可能出现这样的异常

### throw

- 用在方法体内，跟的是异常对象名
- 表示手动抛出异常对象，由方法体内的语句处理

## 抛出处理异常的意义

- ① 在方法中，当传递的参数有误，没有继续运行下去的意义了，则采取抛出处理，表示让该方法结束运行。
- ② 告诉调用者方法中出现了问题。

## 异常处理方式 —— try...catch...

格式:

```
try {  
    可能出现异常的代码;  
} catch (异常类名 变量名) {  
    异常的处理代码;  
}
```

**好处:** 可以让程序继续往下执行。

## 异常处理方式 —— try...catch...

注意：

- ① 如果 try 中没有遇到问题，怎么执行？
- ② 如果 try 中遇到了问题，那么 try 下面的代码还会执行吗？
- ③ 如果出现的问题没有被捕获，那么程序如何运行？
- ④ 同时有可能出现多个异常怎么处理？

## Throwable 的成员方法

方法名	说明
public String getMessage()	返回此 throwable 的详细消息字符串
public String toString()	返回此可抛出的简短描述
public void printStackTrace()	把异常的错误信息输出在控制台



## 两种处理异常方式的小结

### 抛出 throw throws

- ① 在方法中，当传递的参数有误，没有继续运行下去的意义了，则采取抛出处理。表示让该方法结束运行。
- ② 告诉调用者出现了问题。

### 捕获 try...catch

捕获：能让代码继续往下运行。



## 案例：键盘录入数据

需求：

键盘录入学生的姓名和年龄,其中年龄为 18 - 25岁,  
超出这个范围是异常数据不能赋值.需要重新录入,一直录到正确为止。

思路：

- ① 创建学生对象
- ② 键盘录入姓名和年龄，并赋值给学生对象
- ③ 如果是非法数据就再次录入

# 目录 Contents

- ◆ 异常概述
- ◆ 异常处理方式
- ◆ 自定义异常

## 步骤:

- ① 定义异常类
- ② 写继承关系
- ③ 空参构造
- ④ 带参构造

```
public class AgeOutOfBoundsException extends RuntimeException{  
    public AgeOutOfBoundsException() {  
    }  
    public AgeOutOfBoundsException(String message) {  
        super(message);  
    }  
}
```

自定义异常存在的意义：就是为了让控制台的报错信息更加的见名之意。

## 小结

- ① 为什么要学习异常？因为我们要处理代码中出现的问題
- ② 异常的处理方式：抛出和捕获
  - 抛出的意义：让方法停止并告诉调用者这里出现了问题
  - 捕获的意义：让程序可以继续往下执行
- ③ 自定义异常的意义：让程序的提示报错更加的见名知意



传智播客旗下高端IT教育品牌