

# 异常

Java Platform Standard Edition

郑春光

# 课程目标

## CONTENTS

ITEMS **1** 异常的概念

ITEMS **2** 异常的分类

ITEMS **3** 异常的产生

ITEMS **4** 异常的传递

ITEMS **5** 异常的处理

ITEMS **6** 自定义异常

# 什么是异常

- 概念：程序在运行过程中出现的特殊情况。
- 异常处理的必要性：任何程序都可能存在大量的未知问题、错误；如果不对这些问题进行正确处理，则可能导致程序的中断，造成不必要的损失。

Throwable：可抛出的，一切错误或异常的父类，位于java.lang包中。

- |- Error：JVM、硬件、执行逻辑错误，不能手动处理。

- |- Exception：程序在运行和配置中产生的问题，可处理。

  - |- RuntimeException：运行时异常，可处理，可不处理。

  - |- CheckedException：受查异常，必须处理。

- 自动抛出异常：当程序在运行时遇到不符合规范的代码或结果时，会产生异常。
- 手动抛出异常：语法：throw new 异常类型(“实际参数”);
- 产生异常结果：相当于遇到 return 语句，导致程序因异常而终止。

- 异常的传递：按照方法的调用链反向传递，如始终没有处理异常，最终会由JVM进行默认异常处理（打印堆栈跟踪信息）。
- 受查异常：throws 声明异常，修饰在方法参数列表后端。
- 运行时异常：因可处理可不处理，无需声明异常。

```
try {
```

可能出现异常的代码

```
} catch(Exception e) {
```

异常处理的相关代码，如： `getMessage()`、`printStackTrace()`

```
} finally {
```

无论是否出现异常，都需执行的代码结构，常用于释放资源。

```
}
```

- `try{ } catch{ }`
- `try{ } catch{ } catch{ }`
- `try{ } catch{ } finally{ }`
- `try{ } catch{ } catch{ } finally{ }`
- `try{ } finally{ }`
- 注：多重catch，遵循从子（小）到父（大）的顺序，父类异常在最后。



- 需继承自Exception或Exception的子类，常用RuntimeException。
- 必要提供的构造方法：
  - 无参数构造方法
  - String message参数的构造方

- 带有异常声明的方法覆盖：
  - 方法名、参数列表、返回值类型必须和父类相同。
  - 子类的访问修饰符合父类相同或是比父类更宽。
  - 子类中的方法，不能抛出比父类更多、更宽的异常。

- 异常的概念：
  - 程序在运行过程中出现的特殊情况。
- 异常的分类：
  - RuntimeException: 运行时异常, 可处理, 可不处理。
  - CheckedException: 受查异常, 必须处理。
- 异常的产生：
  - `throw new` 异常类型(“实际参数”);
  - 相当于遇到 `return` 语句, 导致程序因异常而终止。
- 异常的传递：
  - 按照方法的调用链反向传递, 如始终没有处理异常, 最终会由JVM进行默认异常处理(打印堆栈跟踪信息)。
- 异常的处理：
  - `try { } catch { } finally { }`
- 带有异常声明的方法覆盖：
  - 子类中的方法, 不能抛出比父类更多、更宽的异常。