



福建師範大學
FUJIAN NORMAL UNIVERSITY

Java语言

学 期 : 2025-2026-1
教 师 : 赵珊珊

地理科学学院、碳中和未来技术学院

SCHOOL OF GEOGRAPHICAL SCIENCES、SCHOOL OF CARBON NEUTRALITY FUTURE TECHNOLOGY



第6章 异常



第6章异常

- 6.1 异常概述
- 6.2 异常处理
- 6.3 Error、Exception和Runtime Exception
- 6.4 自定义异常类



6.1 异常概述

- 异常是一个在程序执行期间发生的事件，它中断了正在执行程序的正常指令流。

```
public class Main {  
    public static void main(String[] args) {  
        int array[] = {0,1,2,3};  
        System.out.println(array[3]/array[0]);  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        int array[] = {0,1,2,3};  
        System.out.println(array[4]/array[0]);  
    }  
}
```

Exception in thread "main" java.lang.ArithmetricException: / by zero
at Main.main(Main.java:4)

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
at Main.main(Main.java:4)



6.2 异常处理

- 6.2.1 try...catch...finally结构
- 其语法格式如下：

```
try{  
    代码  
}catch(异常类型1){  
    代码  
} ....  
catch(异常类型n){  
    代码  
}finally{  
    代码  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        int array[] = {0,1,2,3};  
        try {  
            System.out.println(array[3]/array[0]);  
        }catch(ArithmeticException e) {  
            System.out.println("捕获了运算异常: "+e);  
            return;  
        }finally {  
            System.out.println("执行finally语句块");  
        }  
        System.out.println("捕获异常结束");  
    }  
}
```

捕获了运算异常: java.lang.ArithmetricException: / by zero
执行finally语句块



6.2 异常处理

- 6.2.2 抛出异常
- 若某个方法可能会发生异常，但不想在当前方法中处理这个异常，则可以使用throws或throw关键字在方法中抛出异常。

```
public class Main {  
    public static int div(int a,int b) throws ArithmeticException{  
        return a/b;  
    }  
    public static void main(String[] args) {  
        try {  
            System.out.println(div(10,0));  
        }catch(ArithmeticException e) {  
            System.out.println("捕获了异常: "+e);  
        }  
    }  
}
```

捕获了异常: java.lang.ArithmetricException: / by zero



6.2 异常处理

- 6.2.2 抛出异常
- 1. throws 抛出异常
- 其语法格式如下：

数据类型 方法名(形参列表) throws 异常类列表{

方法体;

}

```
public class Main {  
    public static int div(int a,int b) throws ArithmeticException{  
        return a/b;  
    }  
    public static void main(String[] args) {  
        System.out.println(div(10,0));  
    }  
}
```

Exception in thread "main" java.lang.ArithmetiException: / by zero
at Main.div(Main.java:3)
at Main.main(Main.java:6)



6.2 异常处理

- 6.2.2 抛出异常
- 2.throw抛出异常
- 其语法格式如下：

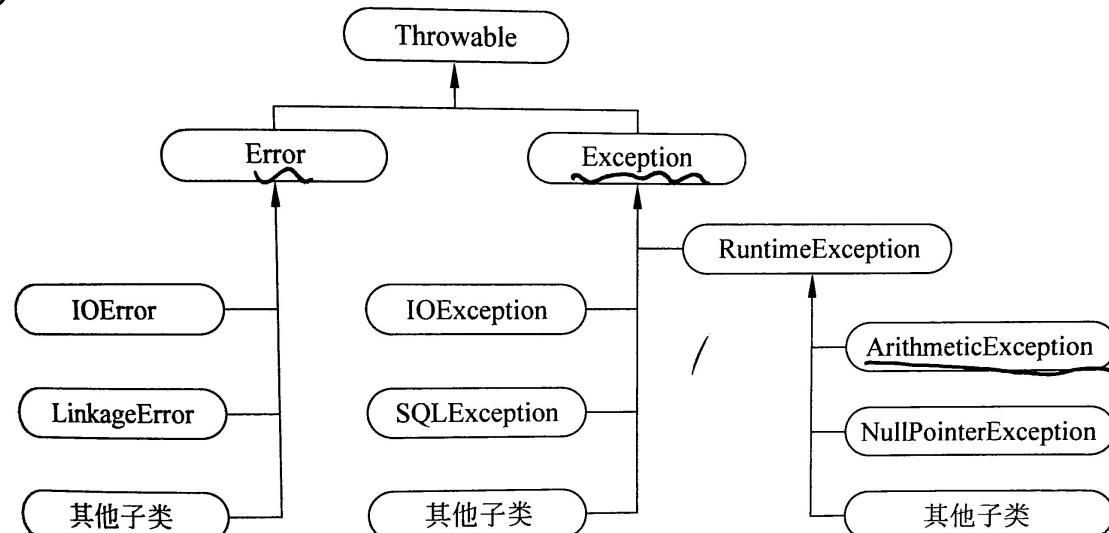
throw new 异常对象();

```
public class Main {  
    public static int div(int a,int b){  
        if(b==0)  
            throw new ArithmeticException("除数不能为0!");  
        return a/b;  
    }  
    public static void main(String[] args) {  
        try {  
            System.out.println(div(10,0));  
        }catch(ArithmeticException e) {  
            System.out.println("捕获了异常: "+e);  
        }  
    }  
}
```

捕获了异常: java.lang.ArithmetricException: 除数不能为0!

6.3 Error、Exception和Runtime Exception

- 异常类都是Throwable类的子类。
- Throwable类派生出Error类和Exception类，**Error类**描述运行系统的内部错误以及资源耗尽的情形，该类错误由Java虚拟机抛出，为免检异常；
- **Exception类**包括**RuntimeException类体系**其他可查异常，可查异常由环境造成，是捕获处理的重点，可以恢复，为必检异常。
- **RuntimeException类体系**包括错误的类型转换、数组越界访问和试图访问空指针等，为免检异常。当RuntimeException出现时，即表示程序员设计的时候出错。





6.4自定义异常类

- 在程序中使用自定义异常类，大体可分为以下几个步骤：
 - 创建自定义异常类并继承Exception基类，如果自定义Runtime异常，则继承RuntimeException基类。
 - 在方法中通过throw关键字抛出异常对象。
 - 如果在当前抛出异常的方法中处理异常，可以使用try-catch语句块捕获并处理，否则在方法的声明处通过throws关键字指明要抛出给方法调用者的异常，继续进行下一步操作。
 - 在出现异常方法的调用者中捕获并处理异常。



6.4自定义异常类

```
class DivException extends Exception {  
    public DivException() {  
        super();  
    }  
    public DivException(String message) {  
        super(message);  
    }  
}  
public class Main {  
    public static int div(int a,int b) throws DivException{  
        if(b==0)  
            throw new DivException("除数不能为0! ");  
        return a/b;  
    }  
    public static void main(String[] args) {  
        try {  
            System.out.println(div(10,0));  
        }catch(DivException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

除数不能为0!