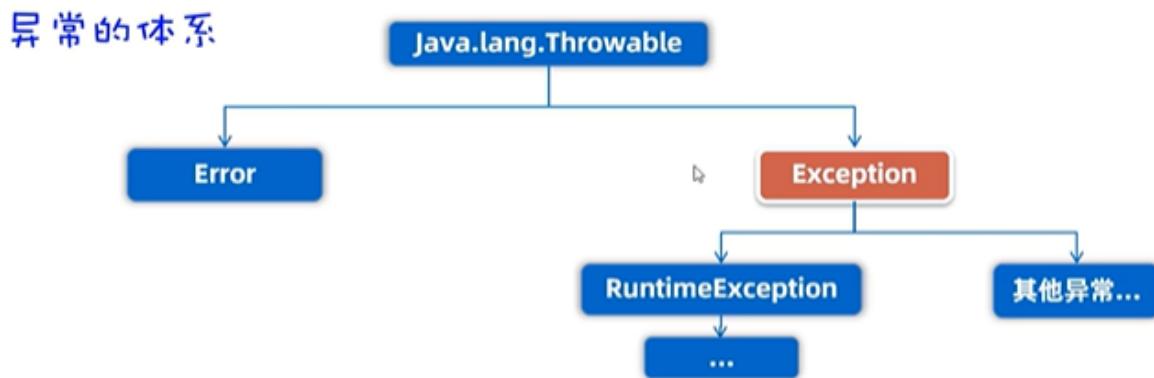


异常

一、异常概述

1.1 异常的体系



- `Error`：系统级别的错误，属于严重问题。
- `Exception`（异常）：即程序可能出现的问题。我们通常使用 `Exception` 及其子类来封装程序可能出现的问题。
 - `RuntimeException`（运行时异常）：编译阶段不会出现错误提醒，只有执行程序时才会报错。比如：数组索引越界异常。
 - 其他异常（也叫编译时异常）：编译阶段就会出现错误提醒。比如：日期时间解析异常。

1.2 异常的作用

- 是检索代码bug的关键参考信息。
- 可以作为方法内部的一种特殊返回值，以便通知上层调用者代码底层的执行情况。

1.3 捕获与抛出异常

P1 捕获异常

- 针对性捕获：

```
1 | try{  
2 |     // 监视可能出现异常的代码  
3 | } catch(XxxException e1){  
4 |     // 处理异常1  
5 | } catch(YyyException e2){  
6 |     // 处理异常2  
7 | }
```

- 全部捕获：

```
1 | try{  
2 |     // 监视可能出现异常的代码  
3 | } catch(Exception e){  
4 |     // 处理全部异常  
5 | }
```

P2 抛出异常，交给上层调用者处理

- 针对性抛出：

```
1 | 方法 throws XxxException, YyyException, ...{  
2 |     ...  
3 | }
```

- 全部抛出：

```
1 | 方法 throws Exception{  
2 |     ...  
3 | }
```

二、自定义异常

2.0 引入背景

```
1 public class ExceptionTest {  
2     public static void main(String[] args) {  
3         // 需求：保存一个合法的年龄  
4         try{  
5             saveAge(160);  
6             System.out.println("Method saveAge executes  
7 successfully.");  
8         }catch (Exception e){  
9             e.printStackTrace();  
10            System.out.println("Method saveAge has bug.");  
11        }  
12    }  
13  
14    private static void saveAge(int age) throws AgeIllegalException{  
15        if(age>0 && age<150){  
16            System.out.println("your age is legal and successfully  
17 saved: "+age);  
18        }else{  
19            System.out.println("Your age is illegal!");  
20            // Todo: 用一个自定义异常对象来封装“输入年龄非法”这个问题，  
21            // 便于后续调试代码时快速定位问题并给出相应的解决方案  
22            throw new 自定义异常类("/age is illegal, your age is "+age);  
23    }  
}
```

2.1 自定义运行时异常

Tip

问题不是很严重，不想在程序员编码时打扰他们时使用。

P1 步骤

1. 定义一个异常类，继承 `RuntimeException`。
2. 重写构造器。
3. 通过 `throw new 异常类(xxx)` 来创建自定义异常对象并抛出给上层调用者。 **(最上层调用者是JVM)**

P2 代码

```
1 public class AgeIllegalRuntimeException extends RuntimeException{  
2     public AgeIllegalRuntimeException() {  
3     }  
4  
5     public AgeIllegalRuntimeException(String message) {  
6         super(message);  
7     }  
8 }
```

P3 控制台输出

```
D:\CS-Softwares\JDK\JDK21\bin\java.exe "-javaagent:D:\CS-Softwares\In  
Your age is illegal!  
Method saveAge has bug.  
AgeIllegalRuntimeException: /age is illegal, your age is 160  
    at ExceptionTest.saveAge(ExceptionTest.java:20)  
    at ExceptionTest.main(ExceptionTest.java:5)  
  
Process finished with exit code 0
```

2.2 自定义编译时异常

⌚ Tip

问题比较严重，想要强烈提醒程序员时使用。

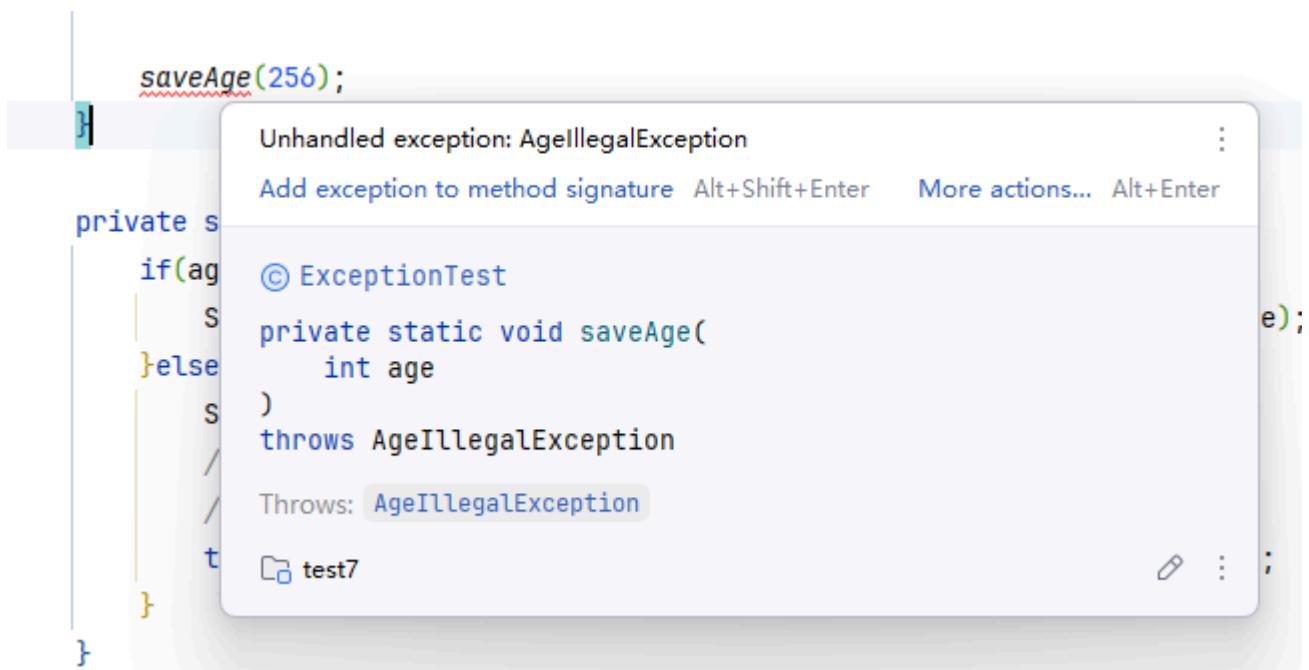
P1 步骤

1. 定义一个异常类，继承 `Exception`。
2. 重写构造器。
3. 通过 `throw new 异常类(xxx)` 来创建自定义异常对象并抛出给上层调用者。

P2 代码

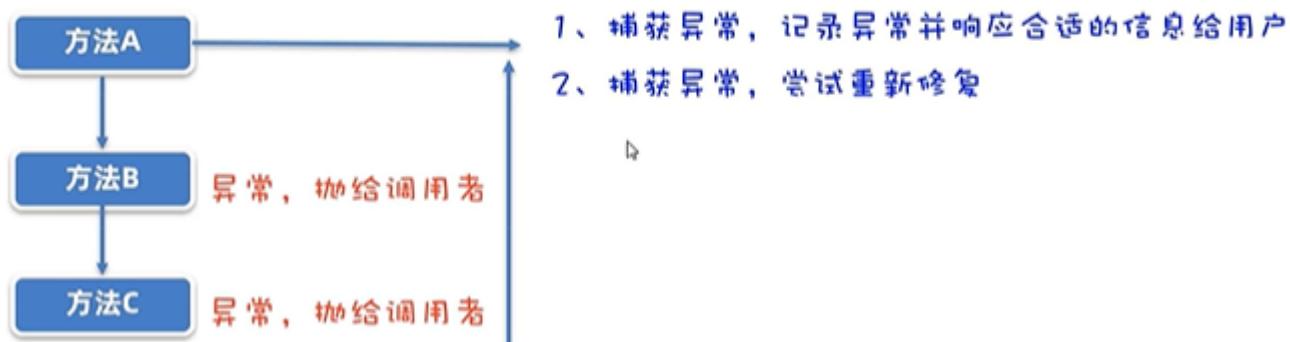
```
1 public class AgeIllegalException extends Exception{  
2     public AgeIllegalException() {  
3     }  
4  
5     public AgeIllegalException(String message) {  
6         super(message);  
7     }  
8 }
```

P3 效果



三、异常的处理

开发中对于异常的常见处理方式



3.1 捕获异常后，记录异常并响应合适的信息给用户

P1 示例代码

```
1 import java.io.FileInputStream;
2 import java.io.FileNotFoundException;
3 import java.io.InputStream;
4 import java.text.ParseException;
5 import java.text.SimpleDateFormat;
6 import java.util.Date;
7
8 public class ExceptionTest {
9     public static void main(String[] args) {
10         // 最上层调用者统一处理异常
11         try {
12             test1();
13         } catch (ParseException e) {
14             System.out.println("The date you would like to parse may
not conform to the format.");
15             e.printStackTrace();
16         } catch (FileNotFoundException e) {
17             System.out.println("The file you're looking for may not
exist.");
18             e.printStackTrace();
19         }
20     }
21
22     private static void test1() throws ParseException,
FileNotFoundException {
```

```
23     SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd  
HH:mm:ss");  
24     Date date = format.parse("2026-11-11 12:37");  
25     System.out.println(date);  
26  
27     test2();  
28 }  
29  
30 private static void test2() throws FileNotFoundException {  
31     InputStream inputStream = new FileInputStream("D:/shift.png");  
32 }  
33 }
```

P2 控制台输出

```
D:\CS-Softwares\JDK\JDK21\bin\java.exe "-javaagent:D:\CS-Softwares\IntelliJ IDEA UL  
The date you would like to parse may not conform to the format.  
java.text.ParseException Create breakpoint : Unparseable date: "2026-11-11 12:37"  
    at java.base/java.text.DateFormat.parse(DateFormat.java:399)  
    at exception_handle.ExceptionTest.test1(ExceptionTest.java:26)  
    at exception_handle.ExceptionTest.main(ExceptionTest.java:14)  
  
Process finished with exit code 0
```

示例代码中，2026-11-11 12:37不符合格式，所以代码执行到此处就报出异常，不会再继续往下执行。

P3 修改为正确日期时间后控制台输出

将示例代码第24行的日期时间改为2026-11-11 12:37:50后，再次运行代码，控制台输出如下：

```
D:\CS-Softwares\JDK\JDK21\bin\java.exe "-javaagent:D:\CS-Softwares\IntelliJ IDEA I
Wed Nov 11 12:37:50 CST 2026
The file you're looking for may not exist.
java.io.FileNotFoundException Create breakpoint : D:\shift.png (系统找不到指定的文件。)
    at java.base/java.io.FileInputStream.open0(Native Method)
    at java.base/java.io.FileInputStream.open(FileInputStream.java:213)
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:152)
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:106)
    at exception_handle.ExceptionTest.test2(ExceptionTest.java:33)
    at exception_handle.ExceptionTest.test1(ExceptionTest.java:29)
    at exception_handle.ExceptionTest.main(ExceptionTest.java:14)

Process finished with exit code 0
```

P4 写法优化

实际开发中，不建议抛出具体的异常，而是直接抛出 `Exception` 这个父类即可。

```
1 import java.io.FileInputStream;
2 import java.io.InputStream;
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5
6 public class ExceptionTest2 {
7     public static void main(String[] args) {
8         // 最上层调用者统一处理异常
9         try {
10             test1();
11         } catch (Exception e) {
12             System.out.println("Your operation has problems.");
13             e.printStackTrace();
14         }
15     }
16
17     private static void test1() throws Exception {
18         SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
19         Date date = format.parse("2026-11-11 12:37:50");
20         System.out.println(date);
21
22         test2();
23     }
24 }
```

```
25     private static void test2() throws Exception {
26         InputStream inputStream = new FileInputStream("D:/shift.png");
27     }
28 }
```

3.2 捕获异常后，尝试重新修复

P1 示例代码

```
1 import java.util.Scanner;
2
3 public class ExceptionTest3 {
4     public static void main(String[] args) {
5         // 需求：调用一个方法，让用户输入一个价格，直到价格合适为止
6         while (true) {
7             try {
8                 System.out.println(getPrice());
9                 break;
10            } catch (Exception e) {
11                System.out.println("Please input a number!");
12            }
13        }
14    }
15
16    private static double getPrice() {
17        Scanner sc = new Scanner(System.in);
18        while (true) {
19            System.out.println("Please input a proper price: ");
20            double price = sc.nextDouble();
21
22            if (price >= 0) {
23                return price;
24            } else {
25                System.out.println("The price you input is not
proper!");
26            }
27        }
28    }
29 }
```

P2 控制台输出

```
D:\CS-Softwares\JDK\JDK21\bin\java.exe "-javaagent:D:\CS-Softwares\AspectJ\lib\aspectjweaver.jar" -Dfile.encoding=UTF-8 com.imooc.aspect.JdkAspect
Please input a proper price:
-98.5
The price you input is not proper!
Please input a proper price:
fsfsaf
Please input a number!
Please input a proper price:
37
37.0

Process finished with exit code 0
```

可以看到，代码具有良好的健壮性。