**异常处理**

程序的容错性

当异常发生时，运行某一部分代码，来减少异常的损失

避免异常 vs 处理异常

异常的分类

Throwable 父类

Error 错误 严重的 不可避免 无法处理

Exception 异常 可以处理

RuntimeException的子类 未检查异常

可以避免 可处理可不处理

非RuntimeException的子类 已检查异常

不可避免 必须处理

如果一个程序可能发生已检查异常，而程序员未加处理，程序的编译出错！

异常对象的产生：

throw：动作 抛出一个异常 将一个异常对象作为方法的返回值返回 等同于return语句

异常对象的传递：

沿着方法调用链反向传递，直至JVM，最终导致程序终止

异常的处理：

1，throws 声明抛出 消极的处理

package chp11;

import java.sql.\*;

import java.io.\*;

public class TestException {

public static void main(String[] args) throws Throwable{

int a=Integer.parseInt(args[0]);

System.out.println("main 1");

ma(a);

System.out.println("main 2");

}

static void ma(int a) throws Exception{

System.out.println("ma 1");

System.out.println("ma 2");

mb(a);

System.out.println("ma 3");

}

static void mb(int a) throws IOException,SQLException{

System.out.println("mb 1");

mc(a);

System.out.println("mb 2");

}

static void mc(int a) throws FileNotFoundException,EOFException,SQLException{

System.out.println("mc 1");

if(a==0)

throw new NullPointerException();

if(a==1)

throw new FileNotFoundException();

if(a==2)

throw new EOFException();

if(a==3)

throw new SQLException();

System.out.println("mc 2");

System.out.println("mc 3");

}

}

2，try-catch 捕获异常 积极的处理

try-catch-finally

finally：无论有没有异常，都会执行 主要用于释放资源

try-finally

自定义异常：

方法的覆盖：访问修饰符相同或者更宽

方法名，参数表，返回值类型相同

子类不能比父类抛出更多的异常