面向过程 面向对象基础 面向对象高级（系统提供的类）

异常处理、集合、多线程、网络通信、IO、反射、常用类(日期、数学、大整数)、图形界面

异常处理

1. 异常的概念和异常处理的优点：

异常和错误的区别？

异常：程序执行过程中出现的问题，该问题可以由程序本身来克服和恢复。出现异常，

则建议修改异常，使正常向下执行。

错误：程序执行过程中出现的问题，该问题不能由程序本身来克服和恢复。出现错误，

则建议程序停止执行。

优点：

提供程序的容错性和健壮性。

将业务处理和异常处理代码分离。

1. 异常类体系：

根：Throwable

Error和Exception

运行时异常：RuntimeException和直接或间接继承了该类的类。

检查异常：除了运行时异常的其他异常类。

要求记忆8种异常：IOException SQLException NullPointerException ClassCastException IndexOutOfBoundsException ArithmeticException FileSystemNotFoundException NumberFormatException

1. 异常处理机制：

try{

业务处理语句组； //try块 对象 instanceof 异常类名

}catch(异常类名1 变量名){

异常处理语句组；

} catch(异常类名2 变量名){

异常处理语句组；

} catch(异常类名3 变量名){

异常处理语句组；

}……

finally{

异常处理语句组；

}

执行过程：

1. 业务处理没有出现异常：

业务处理执行完成后执行finally后的语句。程序接着正常执行。

1. 业务处理出现异常，系统抛出一个异常类对象：
2. 异常类对象被某一个catch捕获，则这些该catch后面的异常处理程序，然后，执行finally后的语句。程序接着正常执行。
3. 异常类对象没有被某一个catch捕获，执行finally后的语句。将异常类对象抛给调用该方法的方法。

注意：

finally中的语句始终都会执行。

try块中还可以再嵌套try-catch-finally

finally和try中的return。

在最后的一个catch中捕获Exception异常。

catch和finally至少要有一个。

练习：

1、从键盘输入输入两个数，输出两个数的商。给三次错误输入机会。如果出错则输出对应的错误信息。如果输入错误超过三次则打印”超出3次机会”，程序执行结束。

2、阅读题：

|  |
| --- |
| public class ThrowTest {  static int a=0;  static void f1(){  try{  a++;  a = f2(a-1);  a++;  return;  }catch (Exception e){  a++;  return;  }finally {  a--;  }  }  static int f2(int a){  return 100/a;  }  public static void main(String[] args) {  f1();  System.out.println(a);  }  } |

1. 抛出异常：

throw 异常类对象；

如果该语句没有在try块中，则该语句后的代码不会执行。异常类对象抛给调用该方法的方法。

如果该语句在try块中，则try块中该语句后的代码不会执行。判断是否能够被try块后面的catch捕获。

方法中如果抛出的异常类对象是检查异常类对象，则需要在方法声明位置(即，大括号和小括号中间的位置)使用throws进行声明或使用catch捕获。

throws 异常类名1，异常类名2，异常类名3，异常类名4……

可能会抛出声明的异常类对象。

在调用声明异常的方法时，需要对声明的检查异常进行处理。

处理方法：使用throws进行声明；使用catch捕获；

throw和throws的区别？

代码位置，方法中，大括号和小括号中间。

一个对象和一个或多个类名

一定抛出异常类对象；可能抛出异常类对象；

throw的应用：

1. 自定义异常类：

class 类名 extends Exception或子类{ //RuntimeException

数据成员；

方法成员；

}

1. 异常类中常用方法：

getMessage

printStackTrace