**异常的基本概念：**

1）在程序运行时产生的一些错误统称为异常。

2）异常是导致程序中断运行的一种指令流。一旦出现异常，如果不进行处理，程序将 退出，后面的语句将不再执行。

3）在Java中所有的异常都是以类和对象的形式存在的。

**异常处理语句：** 异常语句的格式为：  
 **try{**

**可以出现异常的代码段或语句**

**}catch(异常类 异常对象){**

**语句块  
}catch(异常类 异常对象){**

**语句块**

**}finally{**

**一定会执行的代码  
}**

1. try语句块中是可能出现异常的语句或者代码块。
2. catch用来匹配try之中产生的异常对象。
3. finally语句是异常的统一出口，不管是否出现异常，都会执行finally语句块，但finally 语句块可以省略。

**异常类的继承结构：**

在整个Java的异常结构中，Throwable类有两个最常用的子类，即Exception类（异常 类）和Error类（错误类）。

1. Exception类：表示的是程序中出现的异常（所有的异常都是此类的子类）。可以直 接使用try...catch语句处理。
2. Error类：一般情况下指的是JVM错误，在程序中无法对其进行处理。

**异常对象的输出：**

在catch语句块中输出异常的时候，可以使用System.out.println(异常对象)的方式打印异 常信息，也可以通过异常对象调用Exception类中的printStackTrace( )方法输出异常信息。

**异常处理机制：**

Java的异常处理机制也是按照面向对象的方式进行处理的，即：

一旦出现异常，则首先会产生一个异常类的实例化对象，如果存在try语句，则会 将产生异常类对象进行捕捉，然后与catch语句中的各种异常类型进行匹配，如果 匹配成功，则执行catch语句中的代码。

（注：所有异常的对象都可以使用Exception类进行接收，因为Exception类是所有异常 的父类。）

**处理多个异常：**

如果存在多个异常则可以使用多个catch语句进行匹配。但在Java中所有捕获范围小的 异常必须放在捕获范围大的异常之前，即捕捉的范围应由小到大。

**为什么不使用Throwable类接收异常对象：**

对于Throwable类来讲，它包含了Exception类和Error类，而Exception类已经可以接 收所有的异常对象，没有必要再使用Throwable类。

**throws关键字：**

1. 使用throws关键字可以与一个或多个异常类搭配用来声明方法，表示此方法中如果 产生了异常，可以不使用异常语句处理，而是交给方法的调用处进行处理。但是调 用此方法的语句必须使用了try...catch语句处理或者也使用throws声明。
2. 在主方法中也可以使用throws关键字声明，但是主方法是程序的起点，所以主方法 会将异常抛给JVM进行处理。

**throw关键字：**

1）在Java中可以使用throw关键字手工产生一个异常类的实例化对象。

2）如果用户定义了自己的异常类型，这个时候就可以借助throw关键字手工产生一 个异常对象。

**Exception类与RuntimeException类：**

1）RuntimeException类是Exception类的子类；

2）Exception类异常必须使用try...catch语句处理，但RuntimeExcepiton类异常可以不 使用try...catch语句进行处理，由JVM自动处理。但建议在开发中也使用try...catch 语句进行处理。

3）常见的RuntimeException类的子类有：NumberFormatException类、ClassCastException 类、NullPointException类、ArithmeticException类、ArrayIndexOutOfBoundsException 类。

**自定义异常类：**

用户可以根据自己的需要定义自己的异常类，自定义异常类，只需要继承Exception 类或者及其子类即可。

**断言：**

断言就是肯定结果的返回值是正确的，如果是错误的，则输出错误信息，如果没有设置 错误信息，则会输出系统默认的错误信息。使用assert关键字定义断言语句，其格式为：

**assert 表达式;**

**assert 表达式：错误信息;**

1. 断言语句在程序运行时并不会起到任何的作用，如果想要断言语句起作用，则在使 用java命令运行程序时加入参数-ea即可。

**java -ea 类名**

1. 断言语句不能作为条件判断语句出现，在开发中也不建议使用断言。