**异常的基本概念：**

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。

**异常处理：**

在Java中可以使用try…catch…finally语句对异常进行处理。

1. try语句之中是可能出现异常的语句代码，用来捕获产生的异常对象；
2. catch语句用来匹配异常类型并进行处理。在catch语句输出异常时，可以直接使用”System.out.println(异常对象);”的方式打印异常信息，也可以调用Exception类中的printStackTrace( )方法输出异常信息，如下：

异常对象.printStackTrace( );

1. finally语句是异常的统一出口，不管程序是否产生异常，都会执行finally语句，但finally语句可以省略。

**异常类的继承结构：**

在Java的异常结构中，Throwable有两个最常用的类：Exception类（异常类）和Error类（错误类）。

1. Exception类表示程序中出现的异常，可以直接使用try…catch…finally语句处理。
2. Error类表示JVM错误，在程序中无法处理。

**Java的异常处理机制：**

1. 一旦出现异常，则首先会产生一个异常类的实例化对象，在try语句中对产生的异常对象进行捕捉，并与catch语句中的各个异常类型进行匹配，如果匹配成功，则执行catch语句中的代码。
2. 在Java中所有捕获范围小的异常必须放在捕获大的异常之前。
3. Exception类捕获的范围最大，任何异常对象都可以使用Exception类对象接收。

**throws关键字：**

1. 可以使用throws关键字与异常类型搭配声明方法，表示此方法可以不处理异常，如果此方法中出现异常，则自动将产生的异常对象抛给被调用处进行处理，但调用该方法的语句必须也进行了异常的处理。
2. 主方法也可以使用throws关键字声明，但是主方法是程序的起点，所以主方法再向上抛异常时，只能将异常抛给JVM进行处理。主方法中就可以不使用try…catch语句进行异常处理了

**throw关键字：**

1）可以使用throw关键字抛出一个异常。在方法中可以使用throw关键字可以向上抛出方法中已经捕获并匹配了的异常类对象。

2）如果用户自定义了异常类时，系统无法抛出时，可以借助throw手工抛出一个该异常类的对象。

**Exception类与RuntimeException类：**

1. RuntimeException类是Exception类的子类，Exception类异常在程序中必须使用异常处理语句进行处理，而RuntimeException类定义的异常可以不使用异常处理语句进行处理，如果有异常产生，则异常由JVM自动处理，但最好也使用try…catch语句处理。
2. 如果一个方法使用throws关键字搭配的异常类型是RunnableException，则调用此方法的语句可以不使用异常语句处理。
3. 常见的RuntimeException类定义的异常有：NumberformatException（字符串转换为数字异常类）、ClassCastException（类型强制转换异常类）、NullPointerException（空指针异常类）、ArithmeticException（算数异常类）、ArrayIndexOutOfBoundsException（数组下标越界异常类）等。

**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，自定义异常类时，只需要继承Exception类及其子类即可。

**断言：**

断言就是肯定某一结果返回的布尔值是正确的，如果是错误的，则输出错误信息，如果没有设置错误信息，则输出系统默认设置的错误信息。格式如下：

**assert 表达式；** 或  **assert 表达式 ：错误信息；**

1. 程序正常运行时断言并不会起作用，如果想让断言起作用，则在使用java命令运行

程序时加入参数-ea即可

1. 断言不能作为条件判断语句使用
2. 在开发中不提倡使用断言。