1. 异常
2. 定义

程序出现了不正常的情况，程序可以捕获到异常，并且处理这个异常。

1. 异常体系结构

Throwable类是所有错误和异常的父类。错误指的是程序出现的非常严重的问题，程序无法解决。异常是程序可以捕获的轻微的问题，我们可以通过捕获异常解决这个问题。

异常分为 RuntimeException和非RuntimeException。RuntimeException是指在运行阶段才会检查的异常，非RuntimeException是指在编译阶段就会检查的异常。

1. JVM默认处理异常的方式

输出异常名称，错误原因以及异常出现的位置在控制台，并终止程序的执行。

1. 异常的处理
2. try-catch

try {

可能出现异常的代码;

} catch(异常类名 变量名) {

异常的处理代码;

}

程序先进入try代码块中，进行正常逻辑处理，这也是异常可能出现的地方；

当异常出现时，会生成一个异常类对象；

接着，catch中会匹配到该异常类，进入catch代码块中进行异常的处理；

异常处理结束后，程序接着往下执行。

public class ExceptionDemo01 {

  public static void main(String[] args) {

    System.out.println("start");

    method();

    System.out.println("end");

 }

  public static void method() {

    try {

      int[] arr = {1, 2, 3};

      System.out.println(arr[3]);

   } catch (ArrayIndexOutOfBoundsException e) {

//      System.out.println("索引越界");

      e.printStackTrace();

   }

 }

}

1. throws
2. 定义

跟在方法声明的括号后面，表示抛出异常，但只是表示这种异常可能会出现，而异常的处理需要由方法的调用者来处理。

1. 格式

Public void 方法() throws 异常类名{}

1. 示例

public class ExceptionDemo {

  public static void main(String[] args) {

    try {

      method();

   }catch (ParseException e) {

      e.printStackTrace();

   }

 }

  //编译时异常

  public static void method() throws ParseException {

    String s = "2048-08-09";

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Date d = sdf.parse(s);

    System.out.println(d);

 }

}

1. throws 和 throw的区别
2. throw

throw是用在方法体内的，抛出的是异常对象，表示抛出异常，且由方法体内进行处理。当执行throw时说明一定抛出了异常。

1. throws

throws表示抛出异常类名，但只是表示异常可能出现，如果出现，则有方法的调用者来处理异常。

1. 异常的区别
2. 编译时异常

Exception类及其子类，必须显示处理，如用try-catch或者throws抛出异常，否则程序会报错，无法通过编译。

1. 运行时异常

RuntimeException类及其子类，可以不用显示处理。