

异常

一、throw关键字

作用：在用throw关键字指定方法抛出指定的异常

格式：throw new xxxException("产生异常的原因");

注意：

- 1、关键字必须在方法内部
- 2、new后的对象必须是Exception或其子类的对象
- 3、抛出指定异常对象，我们必须处理这个对象

如果是RuntimeException或其子类对象，可以处理

不成熟理解：算是重写了注释？

```
public class demo2 {  
    public static void main(String[] args){  
        int [] a=null;  
        int e=getelement(a,0);  
        System.out.println(e);  
    }  
    public static int getelement(int [] a,int index){  
        if(a==null){  
            throw new NullPointerException("数组是null");  
        }  
        int e=a[index];  
        return e;  
    }  
}
```

二、throws关键字

作用：异常处理的关键字，交给别人处理

格式：

修饰符 返回类型 方法名(参数列表) throws AAAException,BBBException.....{

throws new AAAException("产生原因");

throws new BBBException("产生原因");

.....

}

注意：

- 1、throws关键字必须写在方法声明处
- 2、如果抛出异常，直接声明父类异常

```
package r;

import java.io.FileNotFoundException;
import java.io.IOException;

public class demo2 {
    public static void main(String[] args)throws Exception{
        readfile("d:\\a.txt");
    }
    public static void readfile(String filename) throws Exception{
        if(!filename.equals("c:\\a.txt")){
            throw new FileNotFoundException("传递文件路径不是c:\\a.txt");
        }

        if(!filename.endsWith(".txt")){
            throw new IOException("文件后缀名不对");
        }
        System.out.println("文件读入完成");
    }
}
```

三、try.....catch关键字

格式：try{

可能产生的代码

}catch(定义一个异常变量，用来接收try中抛出的异常对象){

异常的处理逻辑，异常异常对象之后，怎么处理对象

一般在工作中，会把异常的信息记录到一个日志中

}

- 1、catch可有多个

- 2、try产生异常，执行catch的异常处理逻辑，执行完后，继续执行try.....catch之后的算法

```
ackage r;

import java.io.FileNotFoundException;
import java.io.IOException;

public class demo2 {
    public static void main(String[] args)throws Exception{
        try {
            readfile("d:\\a.txt");
        }
    }
}
```

```

    } catch (Exception e) {//try中抛出什么异常对象，catch就定义什么异常变量，用来接收这个异常对象
        // TODO: handle exception
        System.out.println("hhh");
    }
    System.out.println("then");
}
public static void readfile(String filename) throws Exception{
    if(!filename.equals("c:\\a.txt")){
        throw new FileNotFoundException("传递文件路径不是c:\\a.txt");
    }

    if(!filename.endsWith(".txt")){
        throw new IOException("文件后缀名不对");
    }
    System.out.println("文件读入完成");
}
}
}

```

四、finally代码块

格式:

```
try{
```

可能产生的代码

```
}catch(定义一个异常变量，用来接收try中抛出的异常对象){
```

异常的处理逻辑，异常异常对象之后，怎么处理对象

一般在工作中，会把异常的信息记录到一个日志中

```
}finally{
```

无论是否出现异常都会执行

```
}
```

注意:

1.finally不能单独使用，必须和try一起使用

2.一般用于资源释放

```

package r;

import java.io.FileNotFoundException;
import java.io.IOException;

public class demo2 {
    public static void main(String[] args) throws Exception{
        try {
            readfile("d:\\a.txt");
        } catch (Exception e) {//try中抛出什么异常对象，catch就定义什么异常变量，用来接收这个异常对象

```

```

        // TODO: handle exception
        System.out.println("hhh");
    }
    finally{
        System.out.println("资源释放");
    }
    System.out.println("then");
}
public static void readfile(String filename) throws Exception{
    if(!filename.equals("c:\\a.txt")){
        throw new FileNotFoundException("传递文件路径不是c:\\a.txt");
    }

    if(!filename.endsWith(".txt")){
        throw new IOException("文件后缀名不对");
    }
    System.out.println("文件读入完成");
}
}
}

```

五、自定义异常类

格式：

```
public class xxxException extends Exception / RuntimeException{
```

添加一个空参数构造方法

添加一个带异常信息的构造方法

```
}
```

注意：

1、必须继承Exception或RuntimeException

继承Exception：要么throw，要么try.....catch

继承RuntimeException：无需处理，交给虚拟机处理