

# Lab 08 Assignment

班级: 202113

学号: 20373221

姓名: 王宇奇

## Question 1

检查型异常: RuntimeException以外的异常, 从程序语法角度讲是必须进行处理的异常, 如果不处理, 程序就不能编译通过。

非检查型异常: RuntimeException类及其子类异常, 一般是由程序逻辑错误引起的, 程序应该从逻辑角度尽可能避免这类异常的发生。

## Question 2

throws: throws 关键字通常被应用在声明方法时, 用来指定可能抛出的异常。多个异常可以使用逗号隔开。当在主函数中调用该方法时, 如果发生异常, 就会将异常抛给指定异常对象。

throw: throw 关键字通常用在方法体中, 并且抛出一个异常对象。程序在执行到throw语句时立即停止, 它后面的语句都不执行。通过throw抛出异常后, 如果想在上一级代码中来捕获并处理异常, 则需要抛出异常的方法中使用throws关键字在方法声明中指明要跑出的异常; 如果要捕捉throw抛出的异常, 则必须使用try—catch语句。

## Question 3

常见运行时异常: java.lang.NullPointerException, java.lang.ArrayIndexOutOfBoundsException等

常见非运行时异常: java.lang.IOException, java.lang.FileNotFoundException

## Question 4

```
try{
    start();
}
catch (Exception ex){
    ex.printStackTrace();
}
catch (RuntimeException re){
    re.printStackTrace();
}
```

改为

```

try{
    start();
}
catch (RuntimeException re){
    re.printStackTrace();
}
catch (Exception ex){
    ex.printStackTrace();
}

```

将RuntimeException和Exception交换位置，因为RuntimeException是Exception的子类

## Question 5

```

//SuperClass.java
import java.io.IOException;

public class SuperClass {
    public void start() throws IOException{
        throw new IOException("Unable to start");
    }
}

//SubClass.java
import java.io.FileInputStream;

public class SubClass extends SuperClass {
    public void start() throws Exception{
        throw new Exception("Unable to open file");
    }
    public void open(String fileName){
        FileInputStream fis=new FileInputStream(fileName);
    }
}

```

改为

```

//SuperClass.java
import java.io.IOException;

public class SuperClass {
    public void start() throws Exception{
        throw new Exception("Unable to open file");
    }
    public void start() throws IOException{
        throw new IOException("Unable to start");
    }
}

//SubClass.java
import java.io.FileInputStream;

public class SubClass extends SuperClass {
    public void start() throws IOException{
        throw new IOException("Unable to start");
    }
}

```

```
public void open(String fileName){
    FileInputStream fis=new FileInputStream(fileName);
}
}
```

将子类父类的throws异常类互换，子类的异常类不能比父类还大

## Question 6

```
methodA抛出一个异常!
执行methodA的finally!
methodB执行!
执行methodB的finally!
```

## Question 7

```
-----
error
i in finally block:2
1
-----
error
i in finally:okfinally
ok
-----
error
i in finally:okfinally
okfinally
-----
```

如果在执行finally块前出现return语句，会把在值先缓存起来，等执行完finally块后，再返回缓存起来的值。

因此get0和get1中值均没有发生改变，而在 get3 中，i 是一个String builder类型，返回的是一个引用对象。所以后面对 i 修改后，return 的结果也会改变。

## Question 8

见代码