

# Lab 09 Assignment

班级: 212113

学号: 21371220

姓名: 杨硕

## Question1

检查异常：编译器要求你必须处置的异常。代码运行前，编译器就会检查代码会不会出现异常，要求你对可能出现的异常必须做出相应的处理

非检查异常：编译器不要求强制处置的异常，虽然有可能出现错误，但是不会在编译的时候检查，如 `NullPointerException`, `IndexOutOfBoundsException`, `VirtualMachineError`等，这些异常可以在运行时检查或者只能在运行时才能检查出来。

## Question2

throw作用：可以使用throw关键字在指定的方法中抛出指定的异常

throws作用：当方法内部抛出异常对象的时候，我们必须对这个异常进行处理，可以使用throws将异常对象抛出给方法的调用者，最终交给JVM

## Question3

运行时异常：`NullPointerException`、`IndexOutOfBoundsException`

非运行时异常：`IOException`、`FileNotFoundException`

## Question4

```
import java.io.IOException;

public class p04 {
    public static void start() throws IOException, RuntimeException{
        throw new RuntimeException("Unable to start");
    }

    public static void main(String[] args){
        try{
            start();
        }catch (Exception ex){
            ex.printStackTrace();
        }catch (RuntimeException re){
            re.printStackTrace();
        }
    }
}
```

错误：`RuntimeException`类继承自 `Exception` 类，把 `RuntimeException` 放在 `Exception` 之后，`try...catch` 已捕获到 `RuntimeException` 异常错误

修改:

```
//修改1
import java.io.IOException;

public class p04 {
    public static void start() throws IOException, RuntimeException{
        throw new RuntimeException("Unable to Start");
    }

    public static void main(String[] args){
        try{
            start();
        }catch (RuntimeException re){ //移到Exception前面
            re.printStackTrace();
        }catch (Exception ex){
            ex.printStackTrace();
        }
    }
}

//修改2
import java.io.IOException;

public class p04 {
    public static void start() throws IOException, RuntimeException{
        throw new RuntimeException("Unable to Start");
    }

    public static void main(String[] args){
        try{
            start();
        }catch (Exception ex){
            ex.printStackTrace();
        } //删除RuntimeException的catch
    }
}
```

## Question5

```
//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);
    }
}

```

错误：子类 SubClass 重写的 start() 和父类 SuperClass 的 start() 冲突，父类 start() 抛出异常为 IOException，而子类抛出异常为 Exception，IOException 为 Exception 子类；

FileInputStream fis=new FileInputStream(fileName); 未处理异常 FileNotFoundException

修改：

```

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

public class SubClass extends SuperClass {
    public void start() throws IOException{
        throw new IOException("Unable to open file");
    }
    public void open(String fileName){
        try {
            FileInputStream fis=new FileInputStream(fileName);
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
}

```

## Question6

程序输出：

```

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

```

## Question7

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

get0() 中 try 抛出异常，catch 捕获了异常并输出 error，在 return 前先执行了 finally，传给 finally 的 i 的值++，输出为2，因为 i 是基本数据类型，传给finally是值传递，原值不变，返回 i 为1

get1() 中 try 抛出异常，catch 捕获了异常并输出 error，在 return 前先执行了 finally，传给 finally 的 i 的追加 finally，输出为 okfinally，因为 i 是基本数据类型，传给finally是值传递，原值不变，返回 i 为ok

get2() 中 try 抛出异常，catch 捕获了异常并输出 error，在 return 前先执行了 finally，传给 finally 的 i 的追加 finally，输出为 okfinally，因为 i 是引用数据类型，传给finally是引用传递，原值改变，返回 i 为 okfinally

## Question8

程序入口: solution->Question8->lab09Q8->src->Triangle\_Test.java运行