

第五章 异常处理

涉及到课本章节:

- **第5章 异常处理**
- **第8章 输入输出流和文件操作**

异常处理

5.1 Java异常的分类

5.2 Java异常的处理机制

5.3 自定义异常类

异常处理

5.1 Java异常的分类

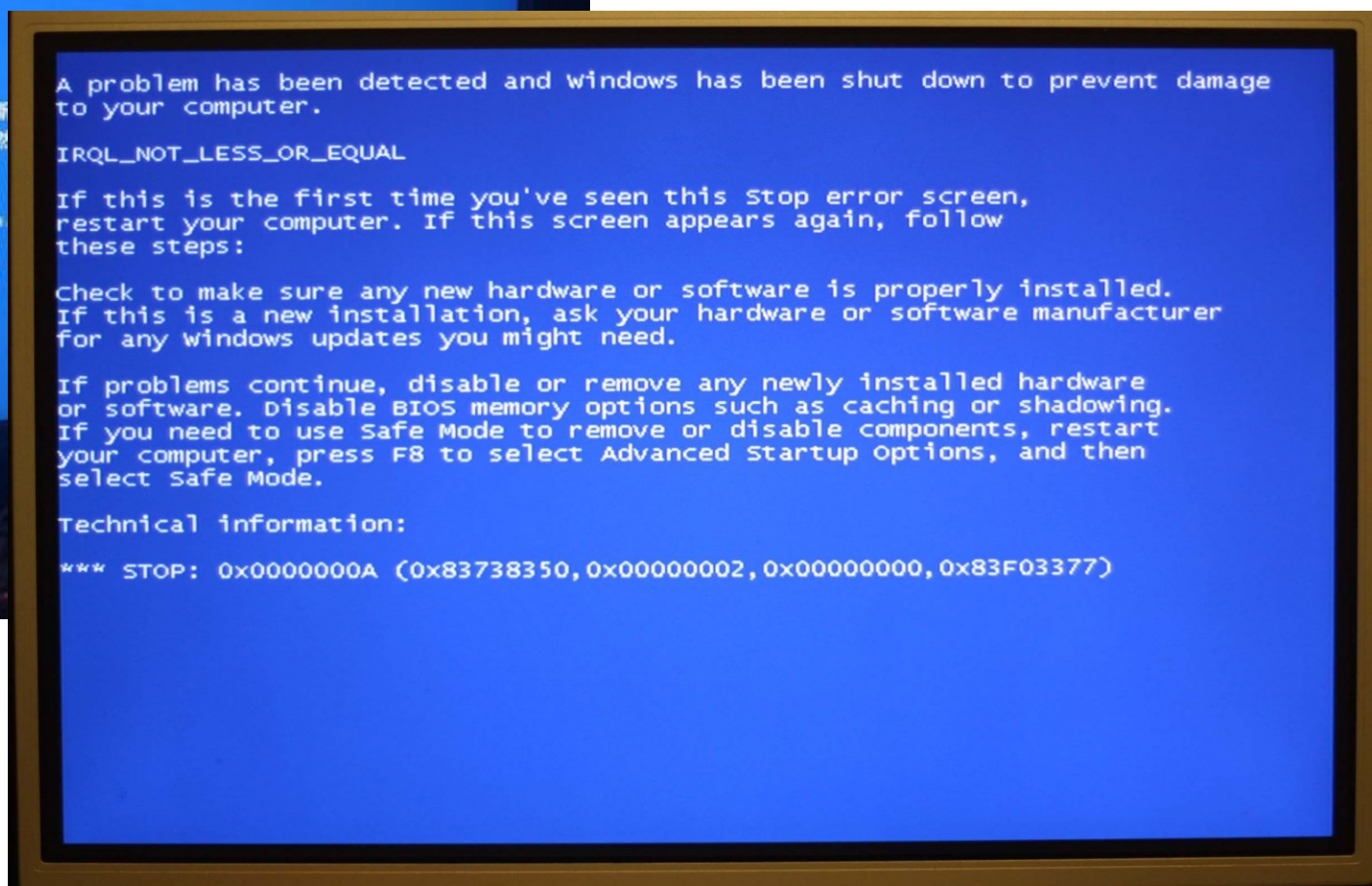
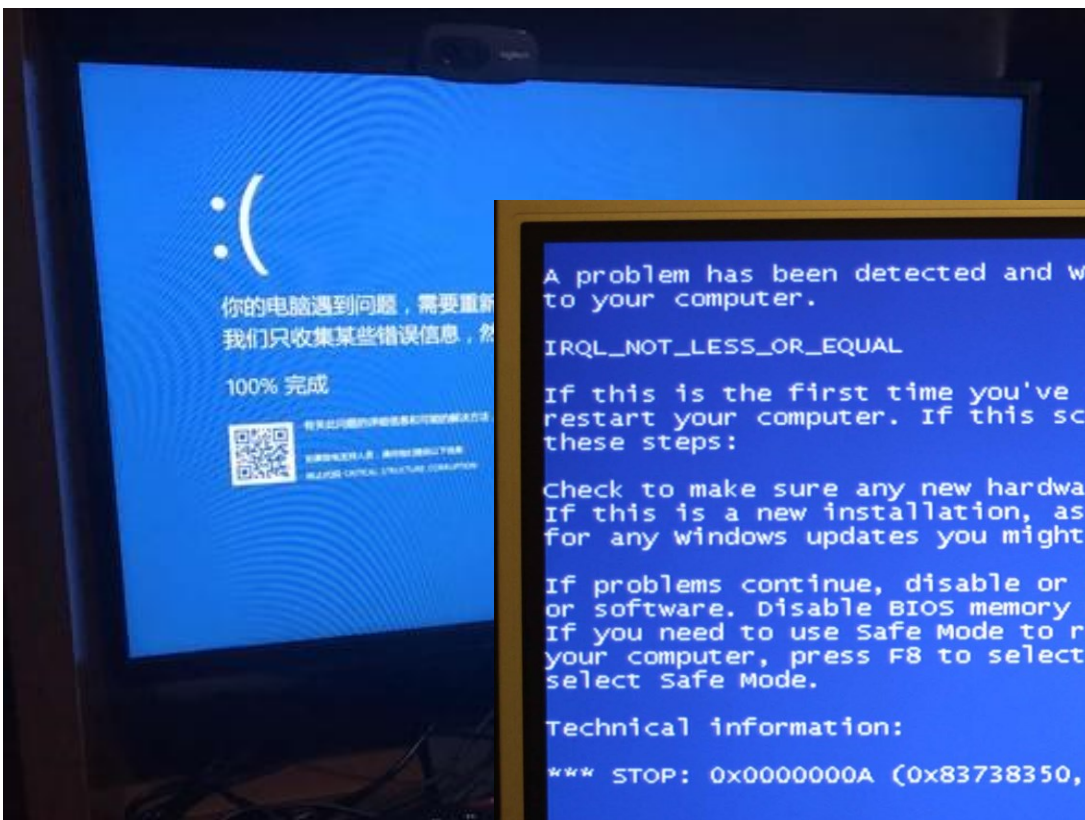
Windows的异常

程序运行前

程序运行中



Windows的极端异常（报错）



序：异常

正常程序流程

```
openTheFile;  
determine its size;  
allocate memory;  
read the file into memory;  
closeTheFile;
```

传统的异常处理方式

```
openTheFile;  
if (theFilesOpen) {  
    determine its size;  
    if (gotTheFileLength){  
        allocate memory;  
        if (gotEnoughMemory) {  
            read the file into memory;  
            if (readFailed) errorCode=-1;  
        }else errorCode=-2;  
    }else errorCode=-3;  
    closeTheFile;  
    if (closeFailed) errorCode=-4;  
}else errorCode=-5;
```

用异常处理机制来捕获处理错误

```
try {  
    openTheFile;  
    determine its size;  
    allocate memory;  
    read the file into memory;  
    closeTheFile;  
}  
catch(fileopenFailed) { dosomething; }  
catch(sizeDetermineFailed) {dosomething;}  
catch(memoryAllocateFailed){ dosomething;}  
catch(readFailed){ dosomething;}  
catch(fileCloseFailed) { dosomething; }
```

异常处理机制的优点(00中)

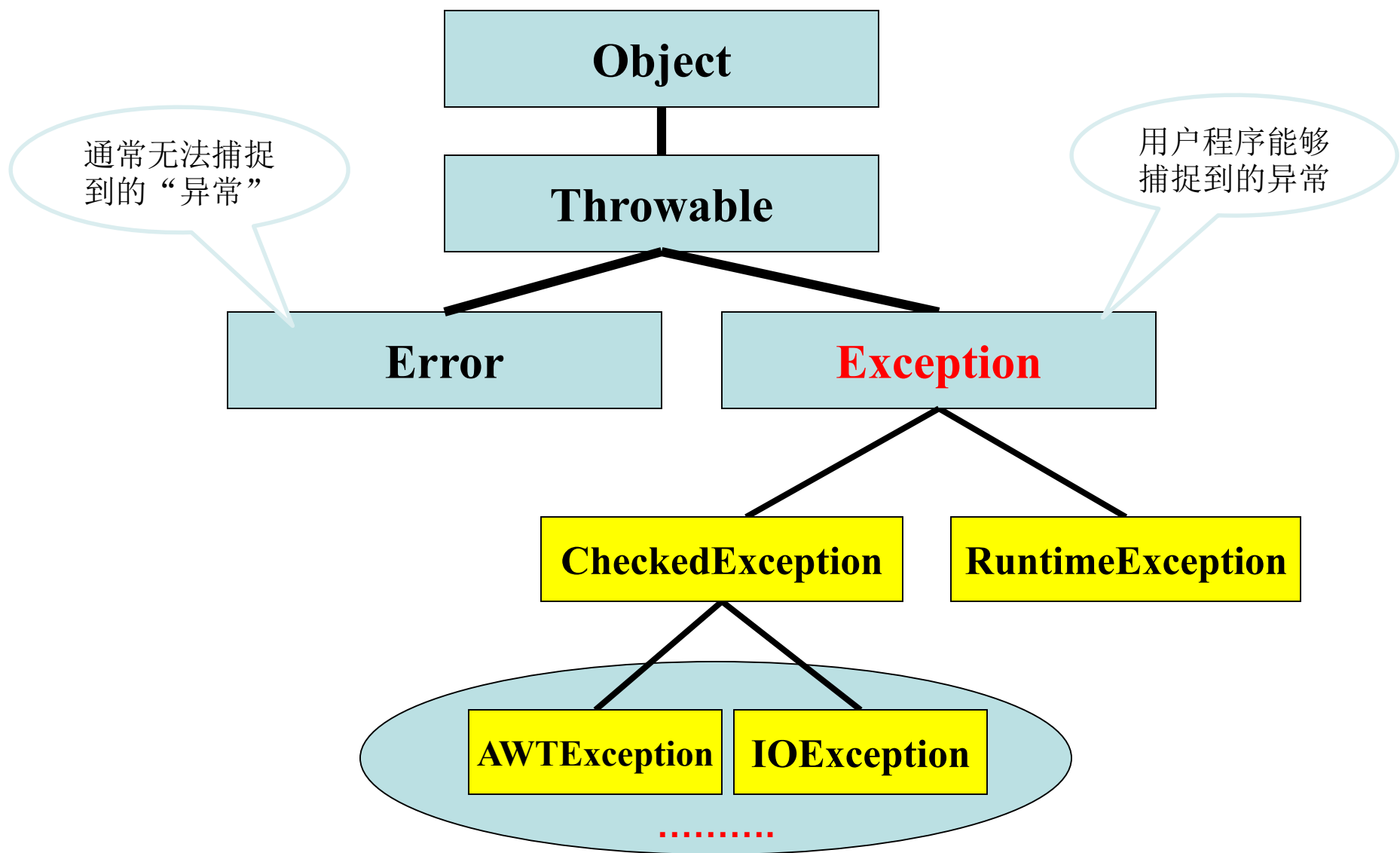
(1) 语法:

异常处理语句将正常代码和“错误”代码分开,结构清晰,重点突出,可读性强.

(2) 运行效果:

使程序具有处理“错误”的能力.即使发生了“错误”,应用程序能够捕获异常并及时处理异常.

5.1 Java异常的分类（Sun定义）



说明:

- **Error** 类对象由 **JVM**生成并抛出给系统，有内存溢出错误、栈溢出错误等通常，**Java**程序不对错误进行处理；
- **Exception** 类对象由应用程序或**JVM**处理抛出的对象。分**RuntimeException**和**checked Exception**。

【类比】

Exception 计划内
Error 计划外

计划去银行取钱，假如没开门就先去超市
没想到路上发生了交通事故

Java Exception:

运行前

checked Exception

运行中

Runtime Exception (unchecked)

ExceptionTypeTest

注: **check**是相对于编译器

【类比】

Exception	计划内	去银行取钱	RuntimeException	超市关门整修
			CheckedException	出门前先检查证件

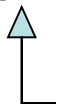
cn.sdu.java.Exception.ExceptionTypeTest

A RuntimeException

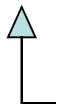
- java.lang.Object



java.lang.Throwable



java.lang.Exception



java.lang.RuntimeException

RuntimeException(46个直接子类)

AnnotationTypeMismatchException, ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, EnumConstantNotPresentException, EventException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IncompleteAnnotationException, IndexOutOfBoundsException, JMRuntimeException, LSEException, MalformedParameterizedTypeException, MirroredTypeException, MirroredTypesException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NoSuchMechanismException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, RejectedExecutionException, SecurityException, SystemException, TypeConstraintException, TypeNotPresentException, UndeclaredThrowableException, UnknownAnnotationValueException, UnknownElementException, UnknownTypeException, UnmodifiableSetException, UnsupportedOperationException, WebServiceException

常见的RuntimeException

- **ArithmeticException**

```
System.out.println(5/0);
```

- **NullPointerException**

```
String s=null;  
boolean eq=s.equals("");
```

- **ArrayIndexOutOfBoundsException**

```
int[] a=new int[3]; int b=a[3];
```

- **StringIndexOutOfBoundsException**

```
String s="hello"; char c=s.charAt(6);
```

常见的RuntimeException

- **ClassCastException**

Object obj=new Object(); String s=(String)obj;

- **IllegalArgumentException**

非法参数，发生转换时经常出现（如日期）

- **NumberFormatException**

继承IllegalArgumentException

int i= Integer.parseInt("ab3");

关于RuntimeException

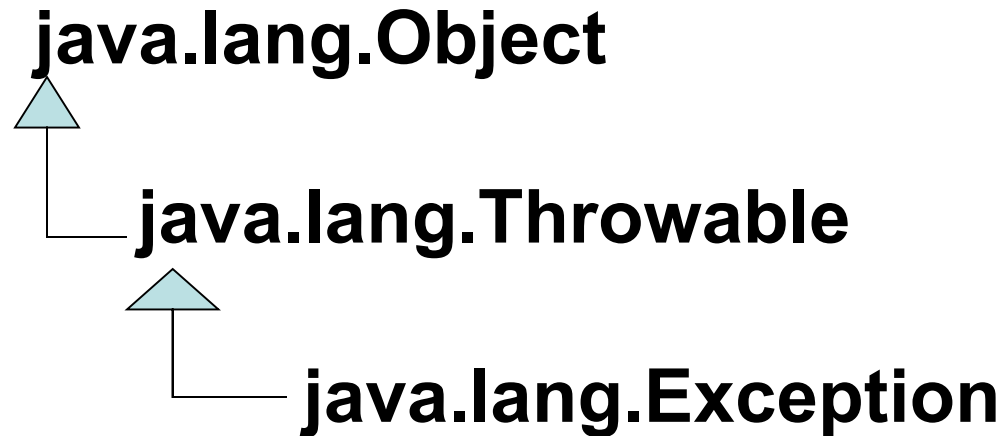
- 属于运行时异常
- 由JVM来接管，程序中可以不用处理

【类比】运行电压不匹配设备异常

民用220V

工业360V 国外110V

B checked Exception



关于checked Exception

- **编译器**要求必须采用**Java**异常处理机制处理
- 常见的有
IO Exception, SQL Exception, AWT Exception

例如，实验中读取txt

总结

- **Checked Exception**必须自己写**try-catch**块做异常处理。
- **RuntimeException**无论是否用**try-catch**捕捉处理都行，区别是不用的话一旦触发将会导致程序运行中断，而用的话则程序不会中断。