# 第一章

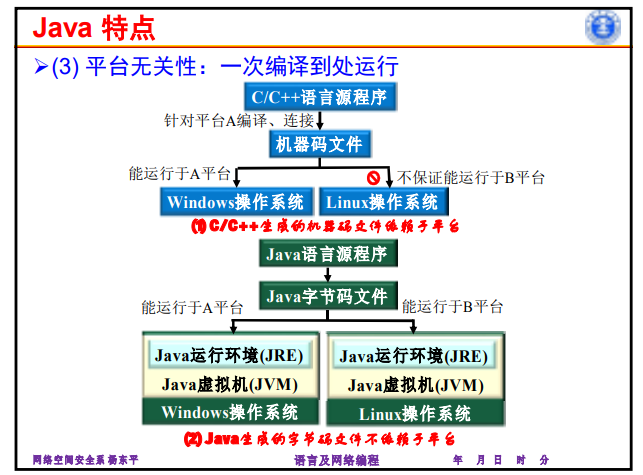
Java语言的的特点：

(1)简单性：语法简单，不使用指针，并提供自动垃圾回收机制，无需担忧内存管理

(2)面向对象

(3)平台无关性：一次编译到处运行

（原理是：编译生成的字节码指令与特定体系结构无关）



(4)安全性：

1.不支持指针，杜绝内存的非法访问

2.自动内存单元收集防止了内存泄露

3. JVM可以自动发现数组和字符串越界，防止堆栈溢出

4.提供异常处理机制，简化错误任务处理

5.要求显示的变量类型声明

6.运行时环境有类装载器，字节码校验器和安全管理器三个组

(5)多线程性：运行环境本身是多线程且语言内置多线程控制

(6)动态性：程序执行时所需要的类在运行时动态加载到内存中，使java程序运行所需的内存开销小

# 第二章

2.1标识符关键字

常量 final

逻辑运算符优先级：

(!) > (&&) > (||)

(!)>算术运算符>关系运算符>(&&) > (||)

位运算符(Bitwise Operators)

按位取反 ~ “~op2”

按位与 & “op1 & op2”

按位或 | “op1 | op2”

按位异或 ^ “op1 ^ op2”

条件运算符

op1 ? op2 : op3

若op1为真，则运算结果为op2，否则为op3

逻辑表达式

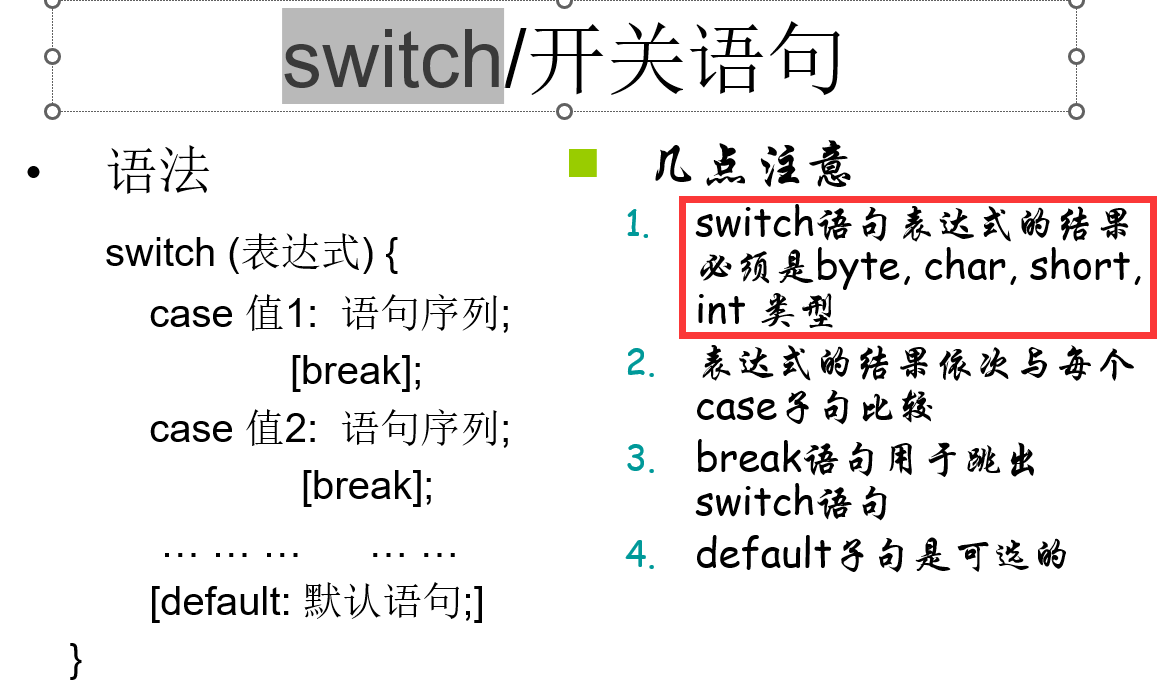
闰年满足下列条件之一：

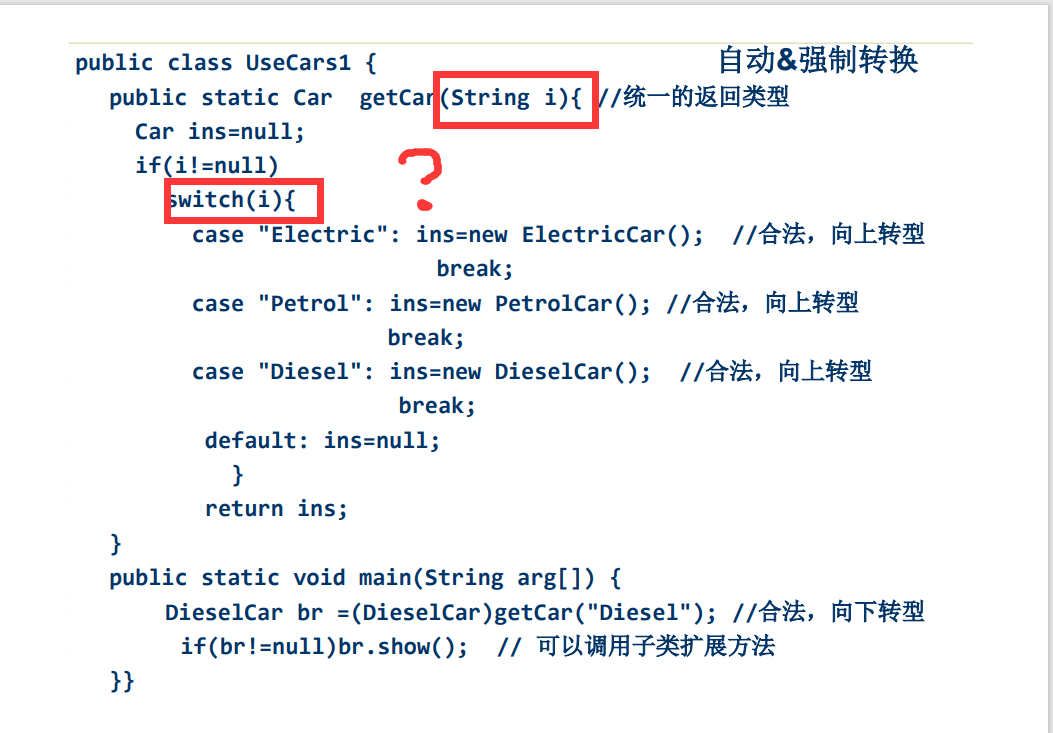
能被4整除，但不能被100整除

能被4整除，又能被400整除

(years%4==0&&years%100!=0)||(years%4==0&&years%400==0)

switch开关语句





当变量表达式所表达的量与其中一个case语句中的常量相符时，就执行此case语句后面的语句，并依次下去执行后面所有case语句中的语句，除非遇到break;语句跳出switch语句为止。

数组

1.初始化

一维数组的初始化

为数组元素指定初始值

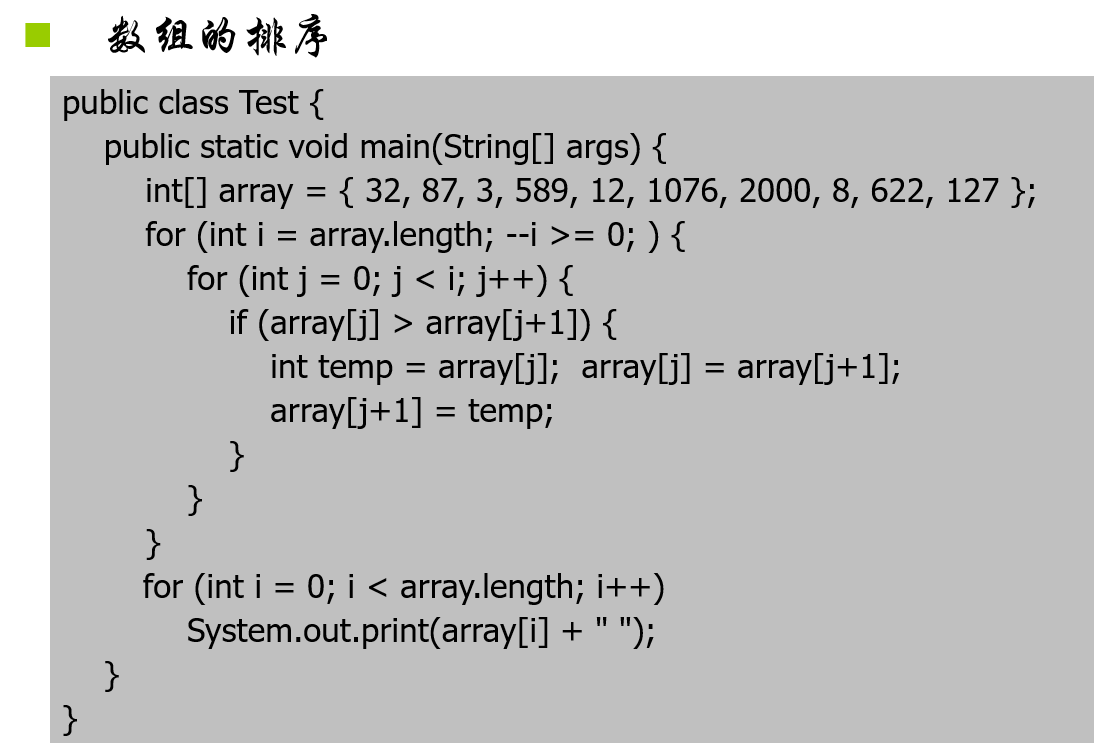
方式一: 声明和创建数组后对数组初始化

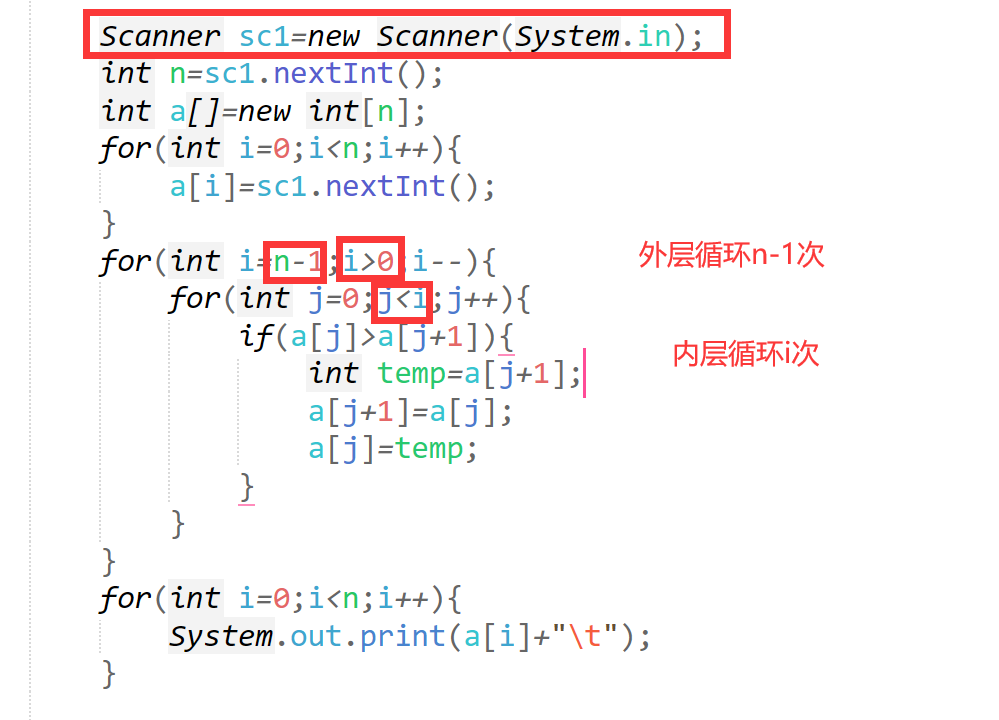
int a[] = new int[5];

方式二: 在声明数组的同时对数组初始化

int a[] = {1, 2, 3, 4, 5};

2.排序

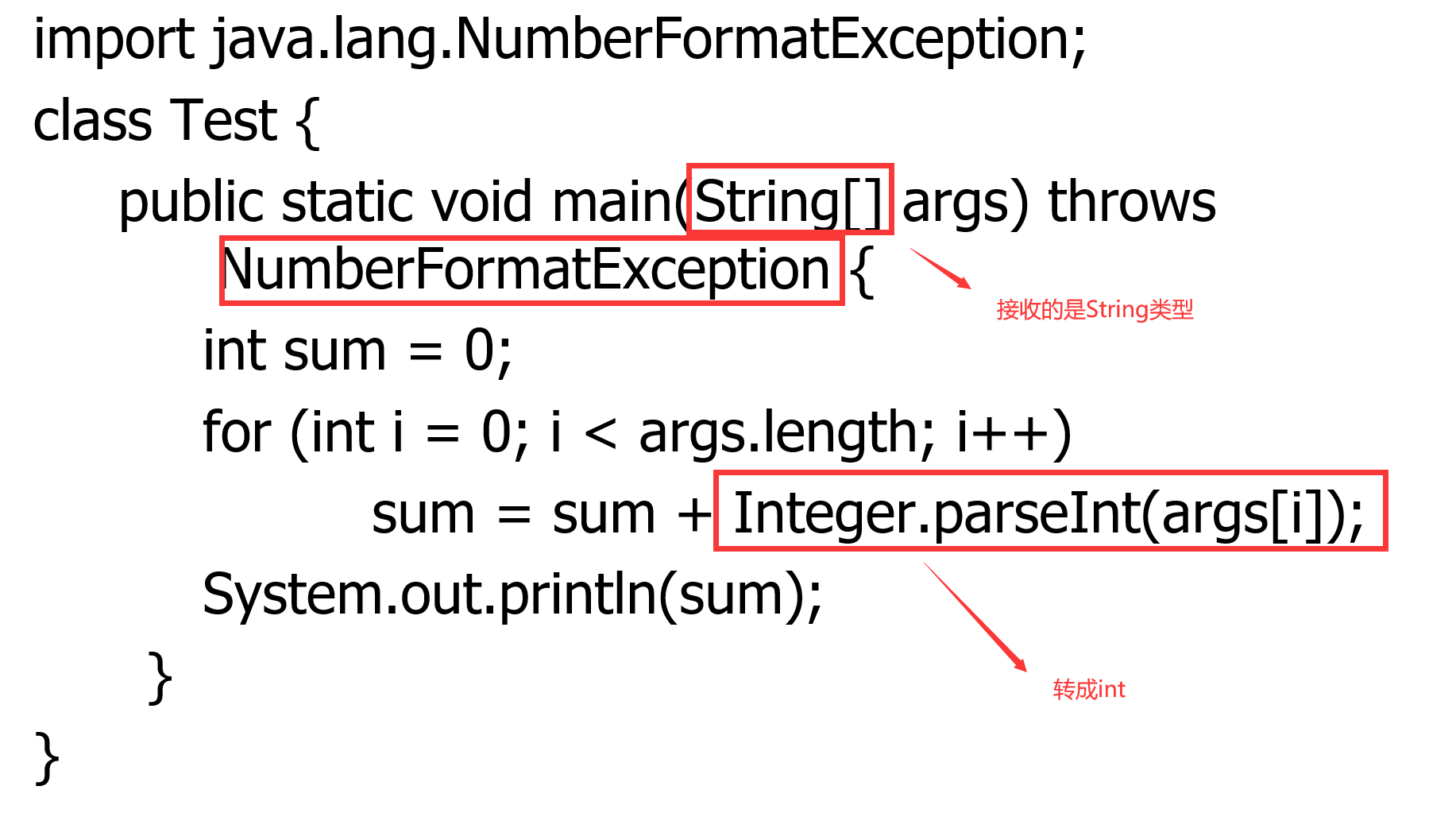




命令行参数：

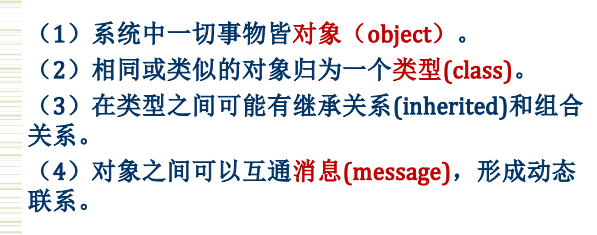
String[] args 是接收命令行传参的数组

args.length args[i]

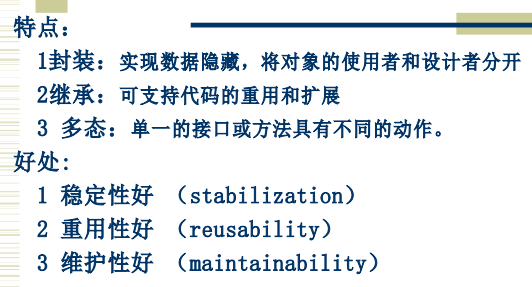


# 第三章 面向对象

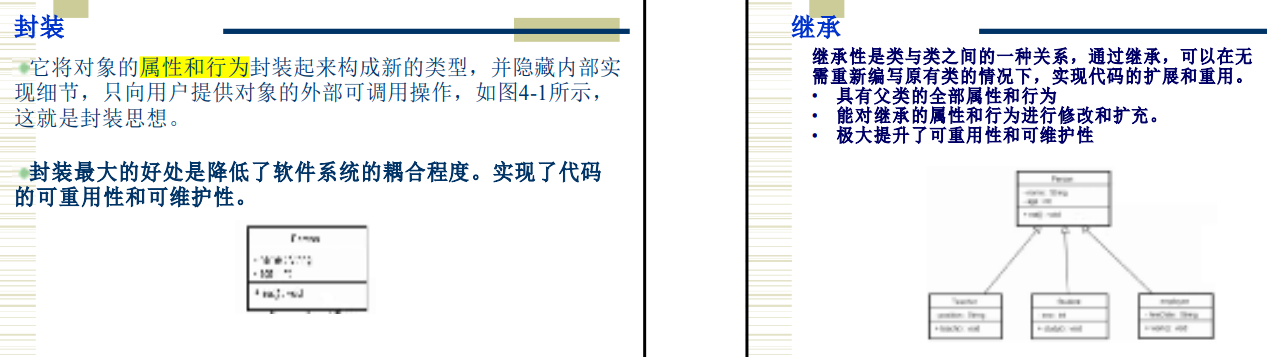
1.面向对象概念



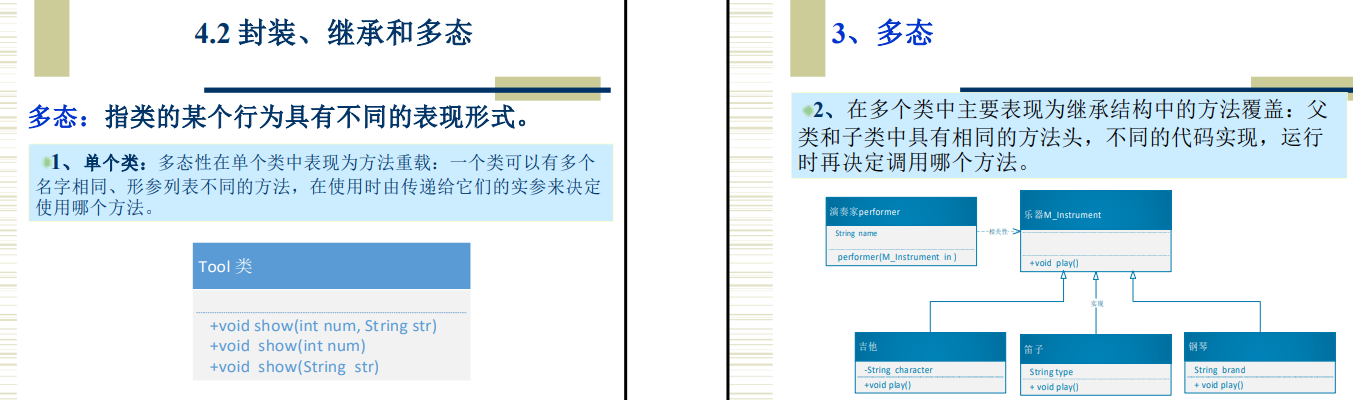
2.面向对象编程的主要特征



封装—》封装成类



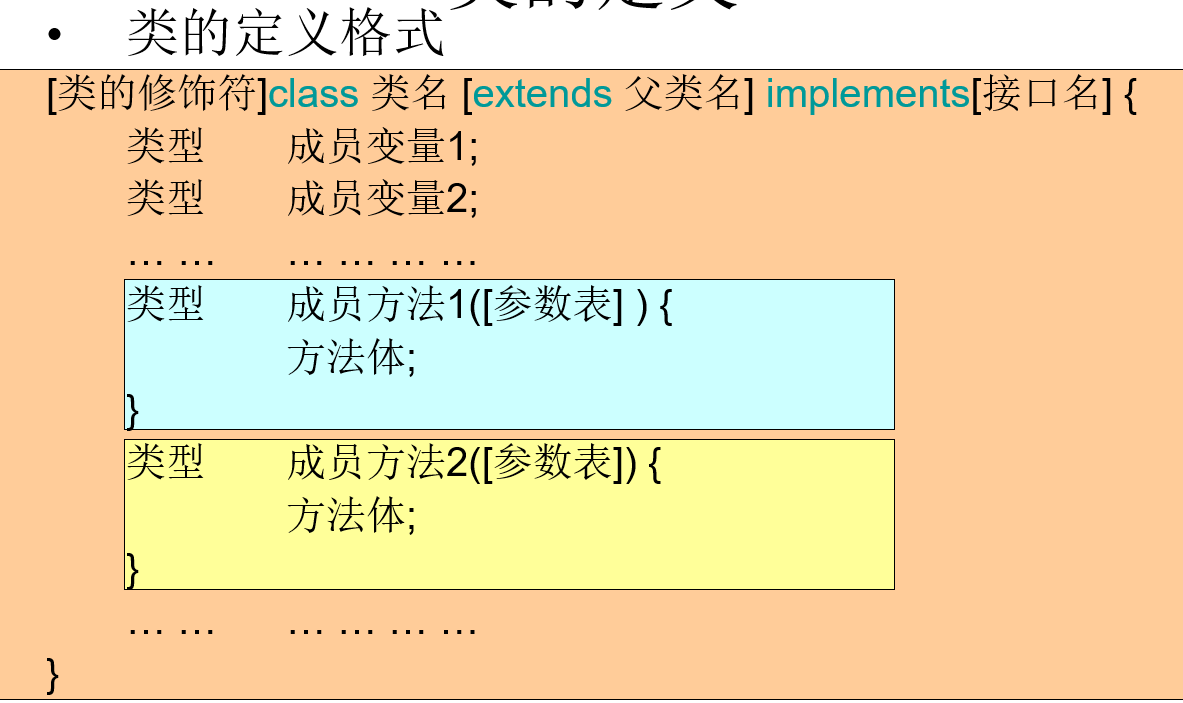
继承中 父类和子类，子类继承(拥有)父类所有的数据和方法，同时子类可以有新的数据和方法



表现在一个类中方法的重载(override/overload)

表现在继承中🡪方法的重写

3.类的定义



4.类修饰符

public: 公共类，可以被其他任何类所使用

无修饰/默认：仅能被同一个包中的其他类引用

abstract: 抽象类，不能被实例化/即创建对象

final: 最终类不能被继承

extends: 一个类继承另一个类

implements: 一个类实现一些接口(interface)的方法，用以变相实现多重继承

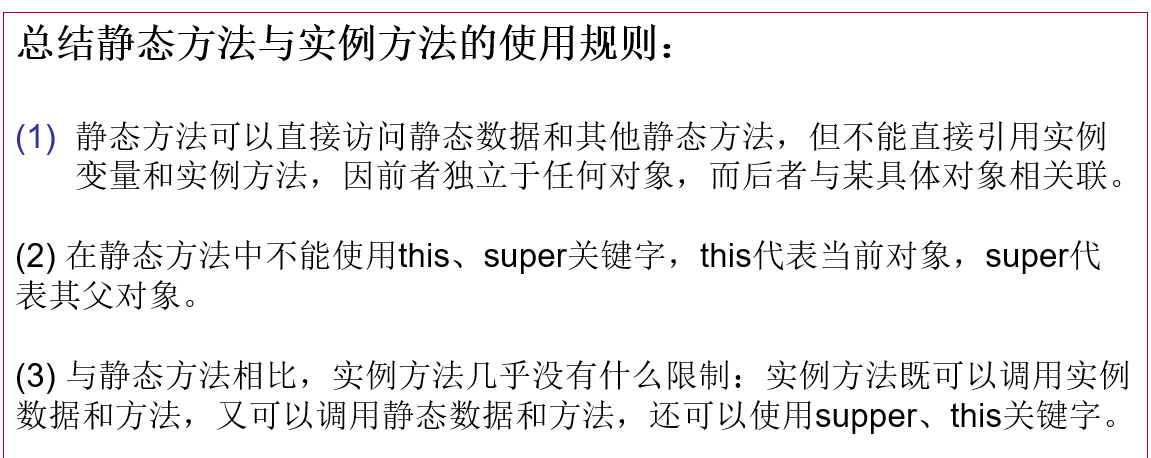
5.类成员的访问



静态成员static

直接通过类(而不必创建对象)就能使用该成员。

-》static方法中仅仅可以调用其他static方法或者创建类对象，通过对象访问非static方法



关键是：静态方法和变量独立与任何对象，且只有一份

如果·想使用this的话，可以转化成类名.变量名的形式使用

最终成员final

值在程序的执行过程中不会改变的变量，或不允许被子类重写的方法

总结：

final->完美，常量，不被继承，不被扩充

static->静态，唯一，类名访问

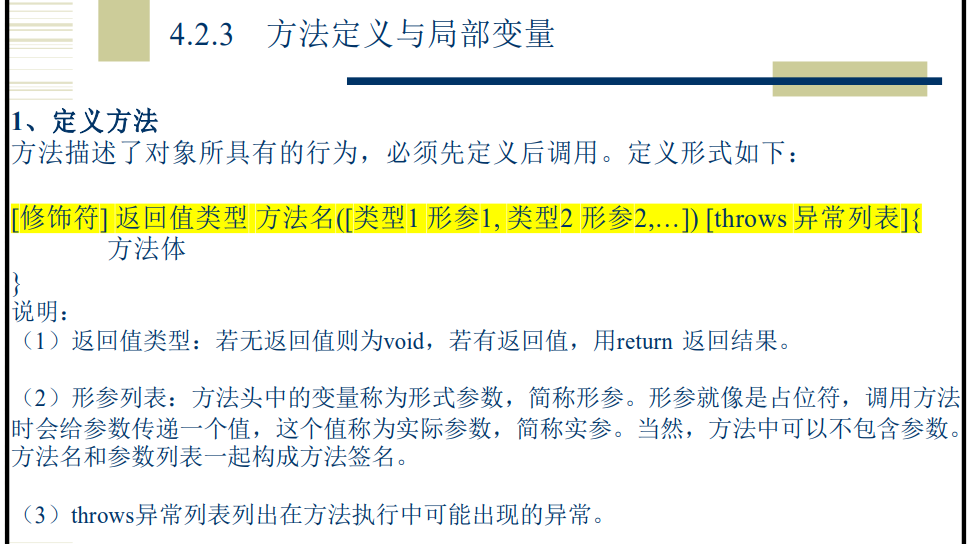
public->所有类可访问

private->类自身访问

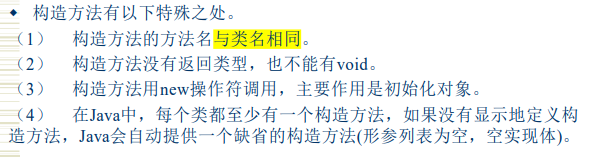
protected->同一包和其子类

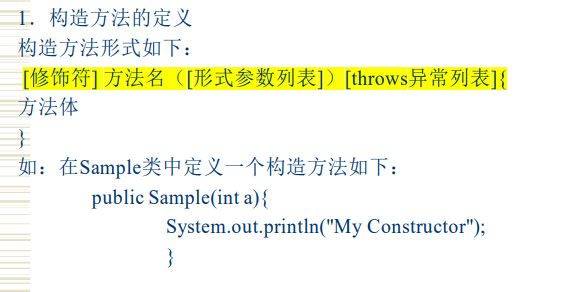
无修饰->同一包

4.方法定义



5.构造方法





6.方法的重载

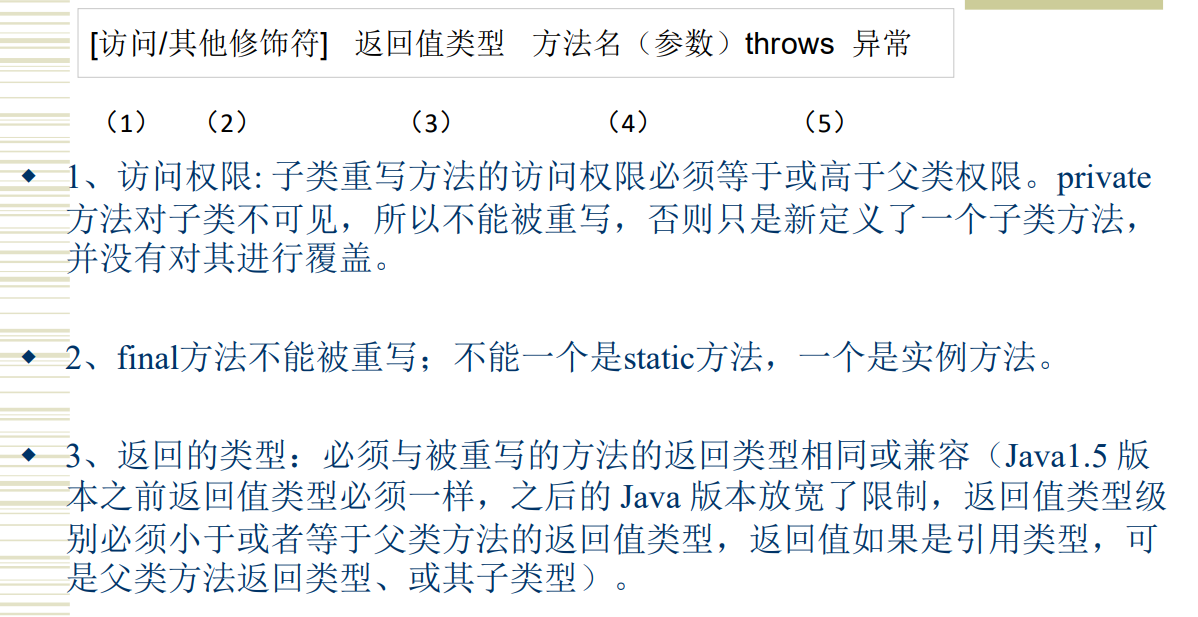
相同方法名，不同的形参列表

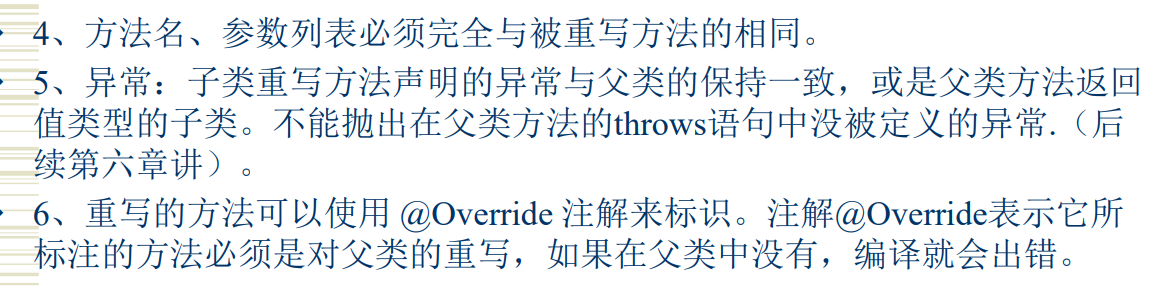
7.方法的重写

重写: 子类和父类的方法名、返回类型和参数均相同（方法可见性可以不同）

final/private方法不能被重写

子类中重写的方法不能降低父类中被重写的方法的可见性

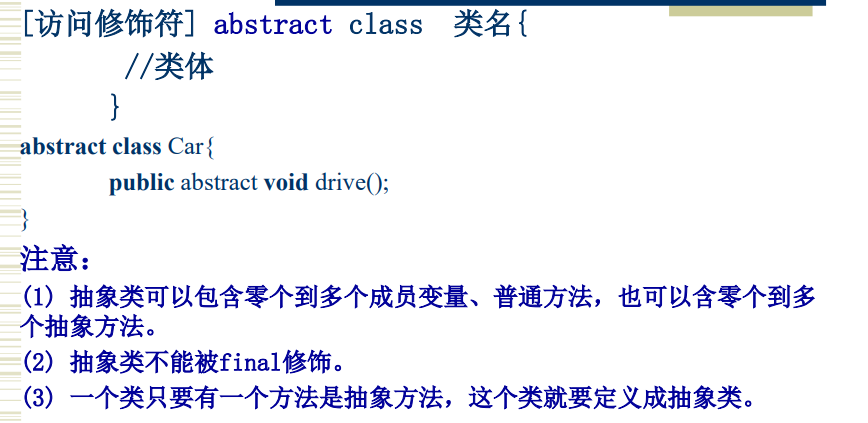




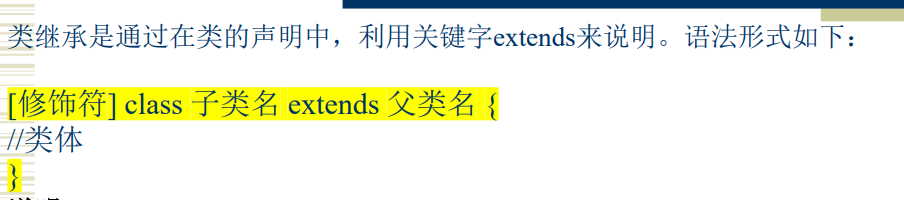
8.抽象类

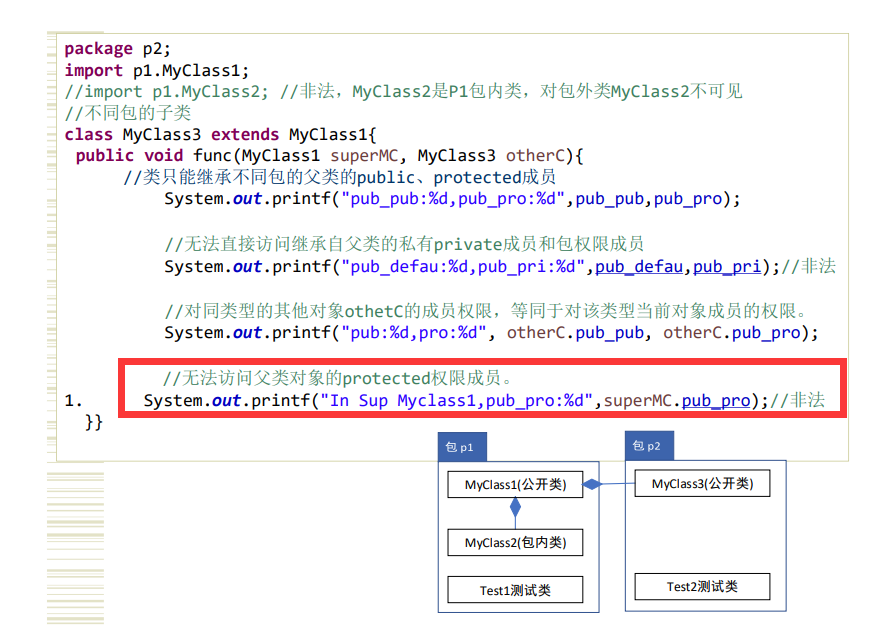
-》未完成的类

抽象方法: 仅仅写出了方法的修饰符、名字、参数、返回类型等，没有编写方法体（即没有方法实现

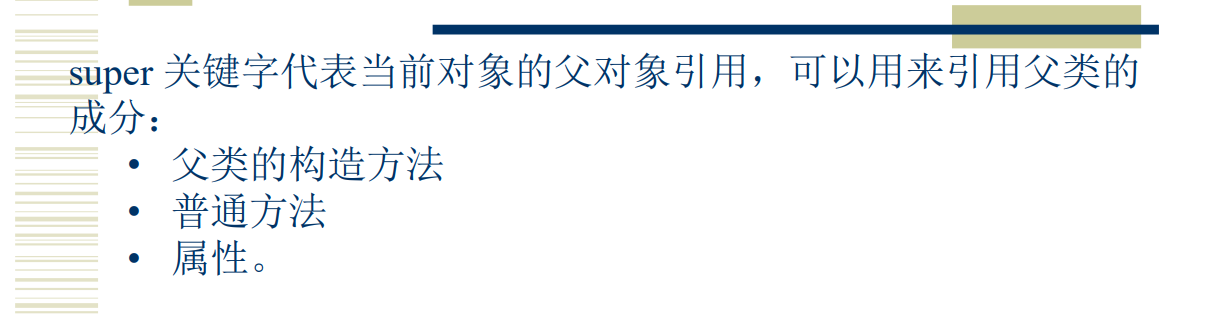


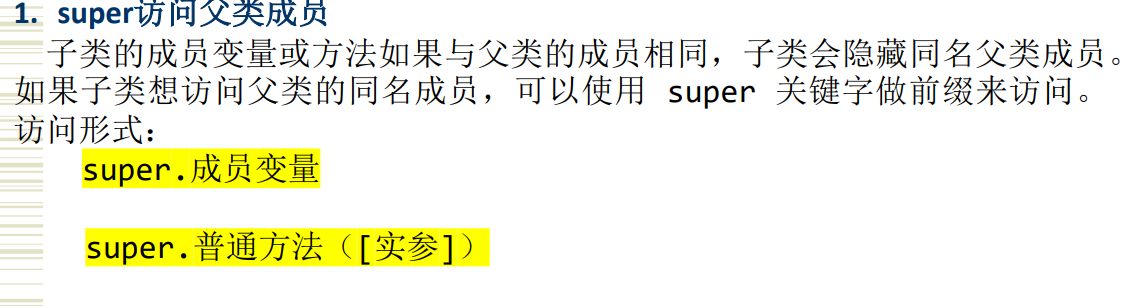
9.继承

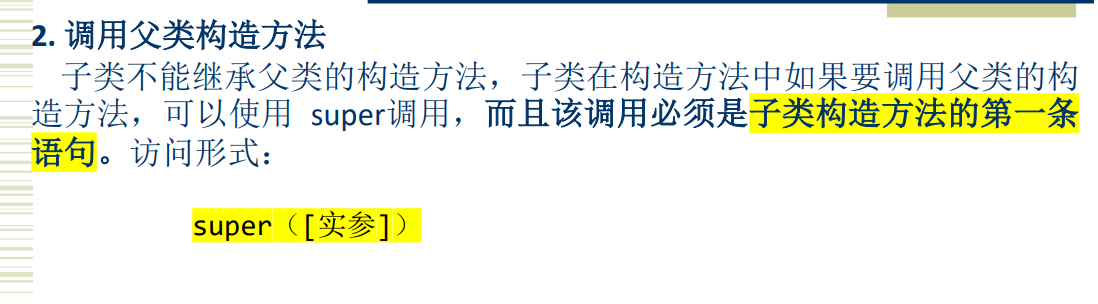


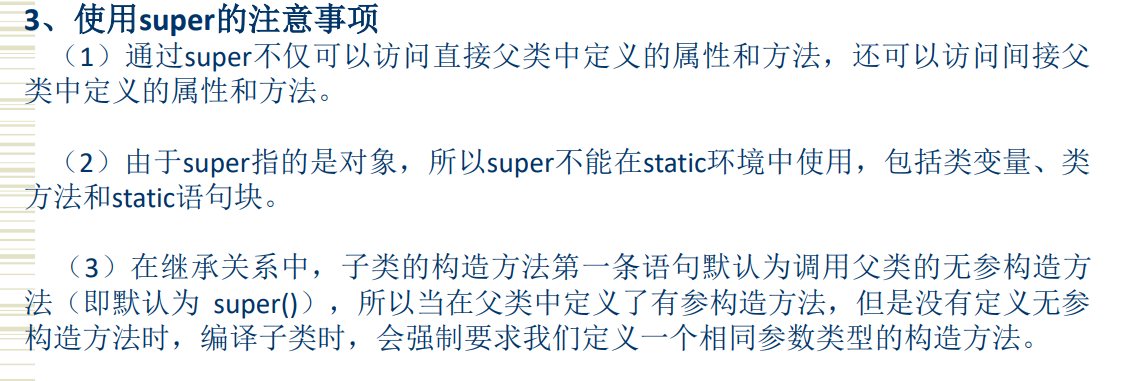


10.super关键字

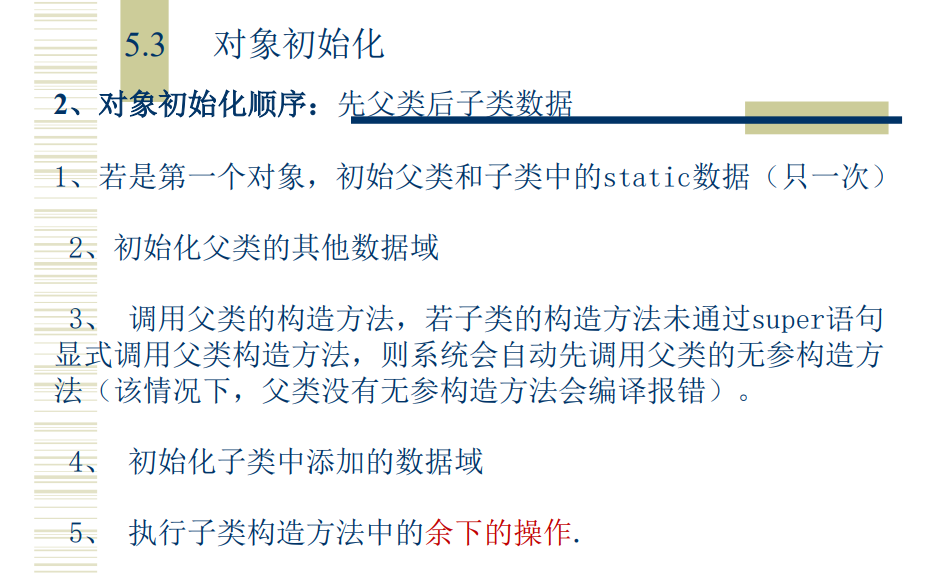




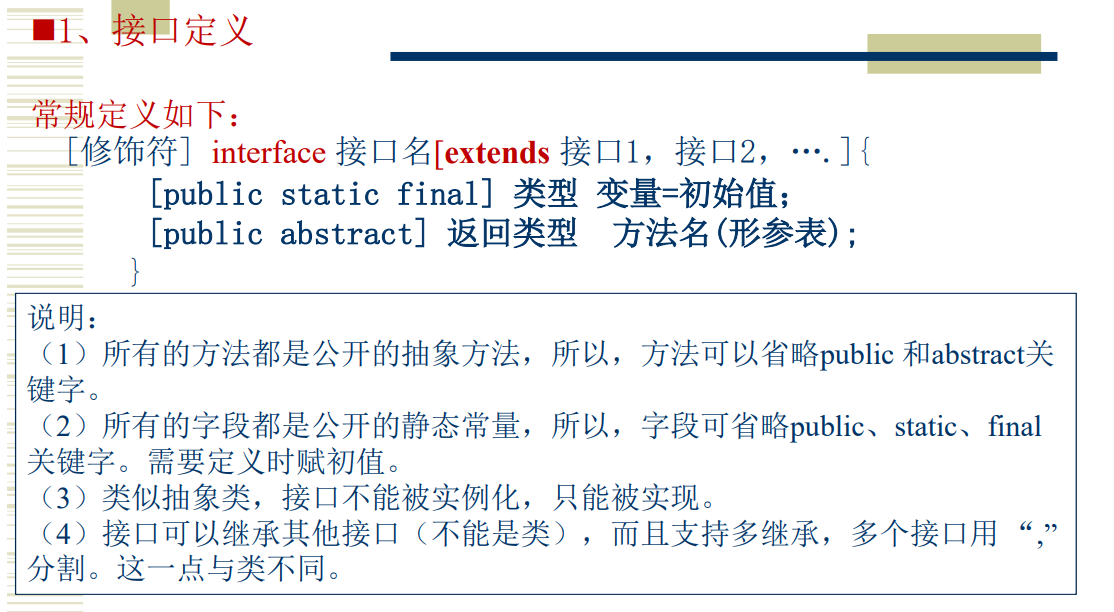


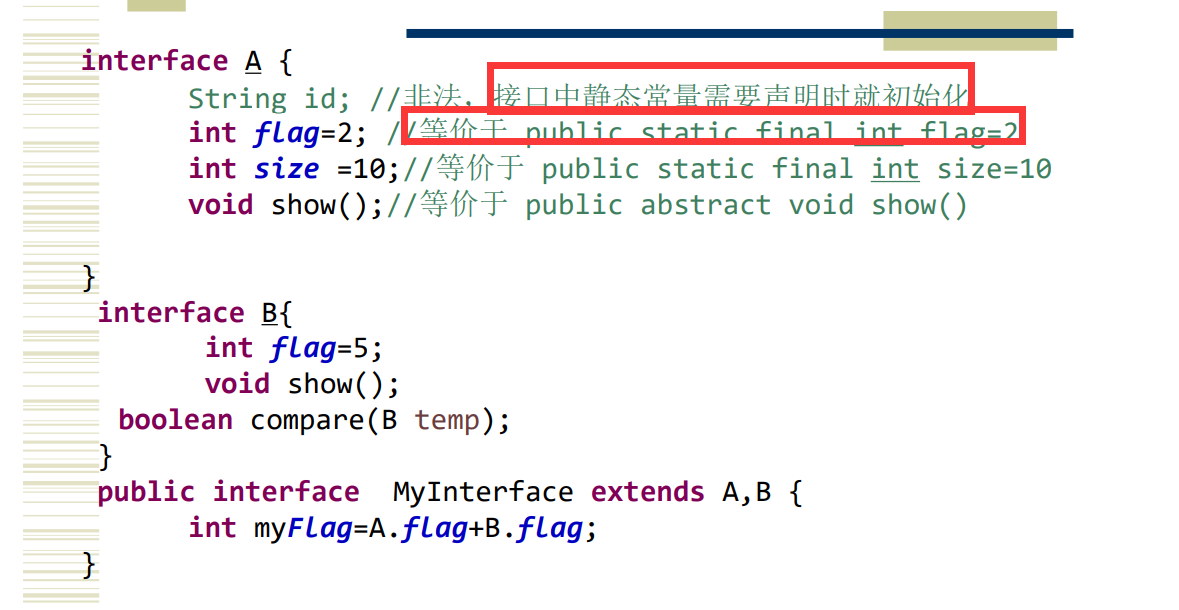


11 对象初始化顺序

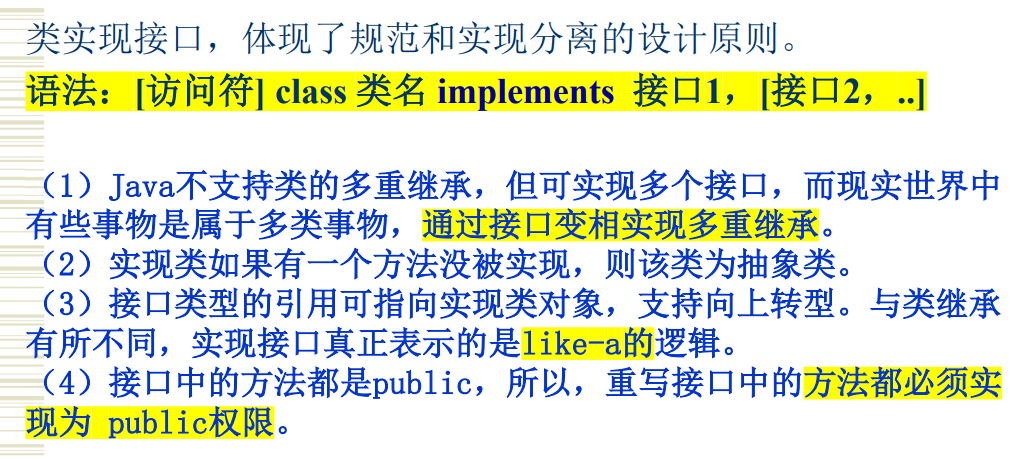


11.接口





12。接口实现

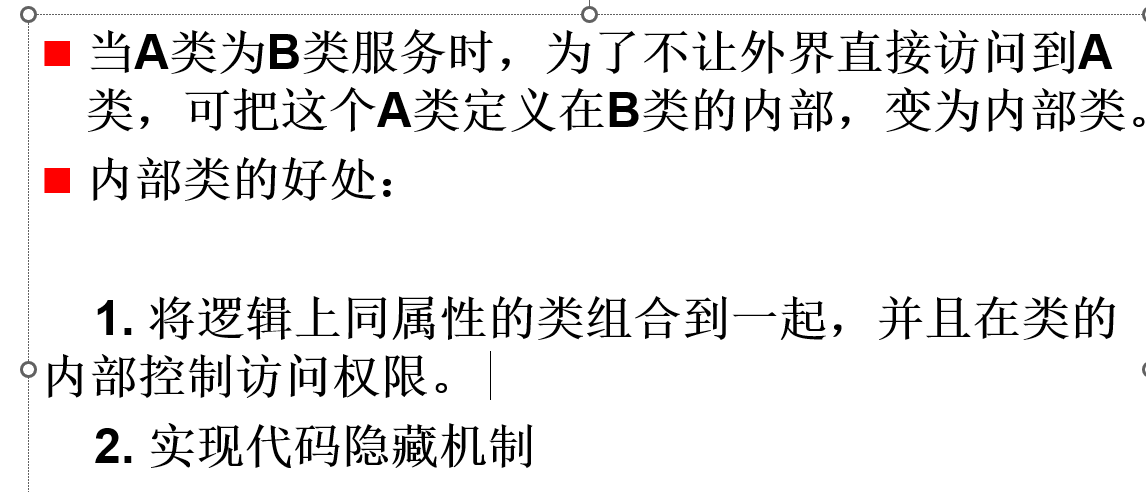


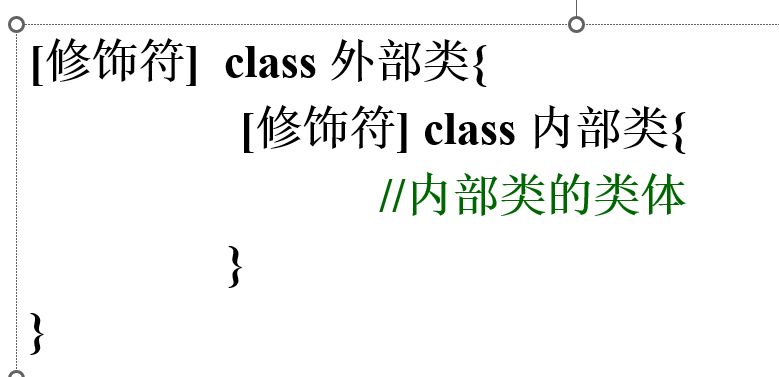
第四点很重要

13.String类型

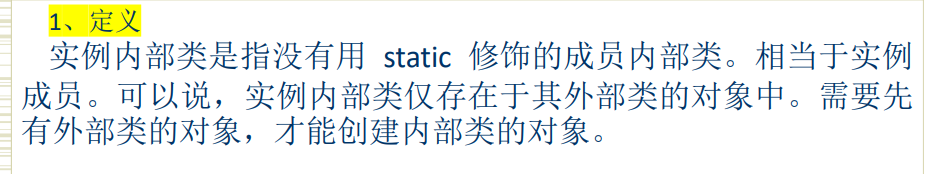
14.比较

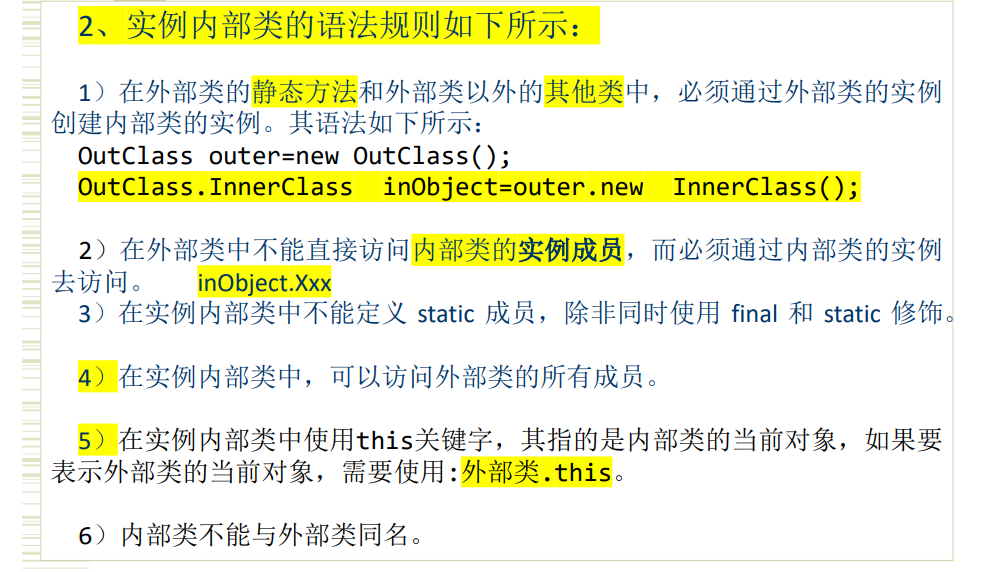
# 第四章 java内部类

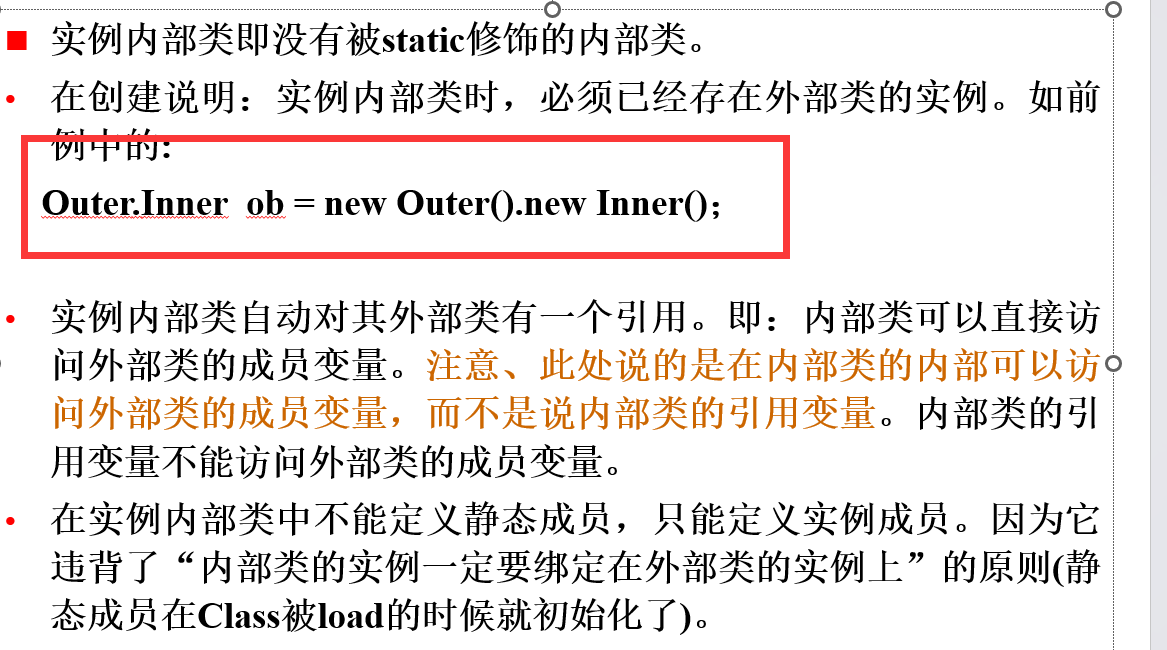




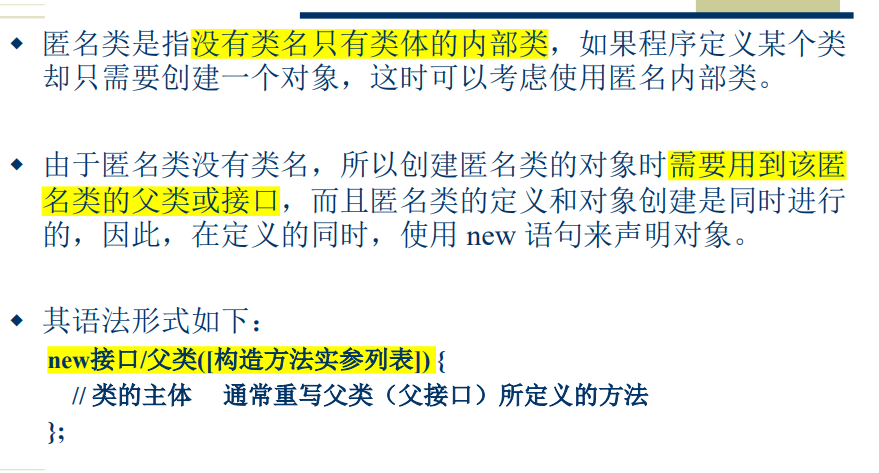
1.实例内部类

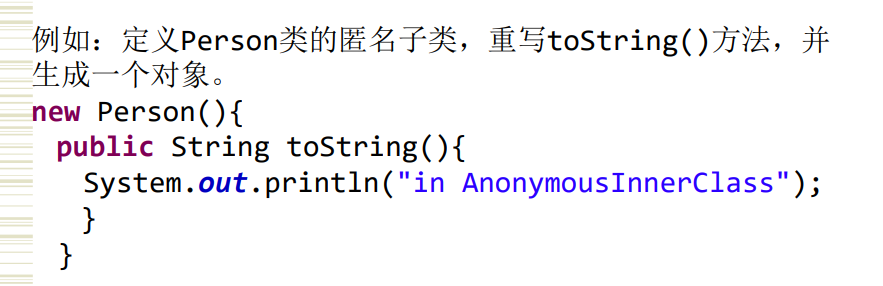


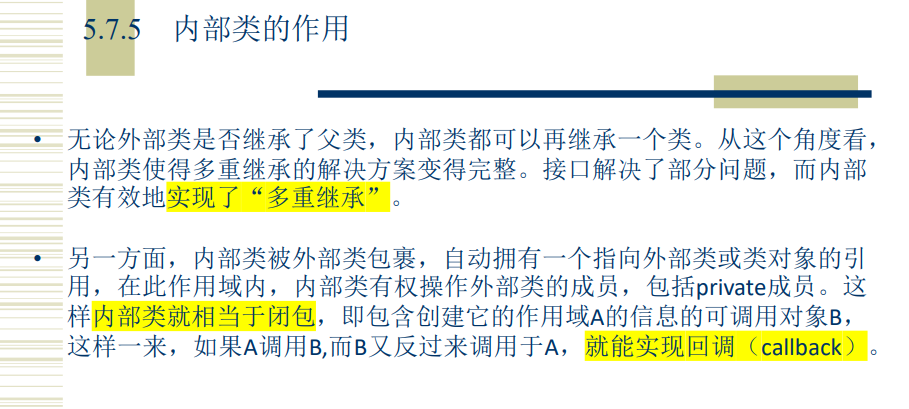




2.匿名内部类



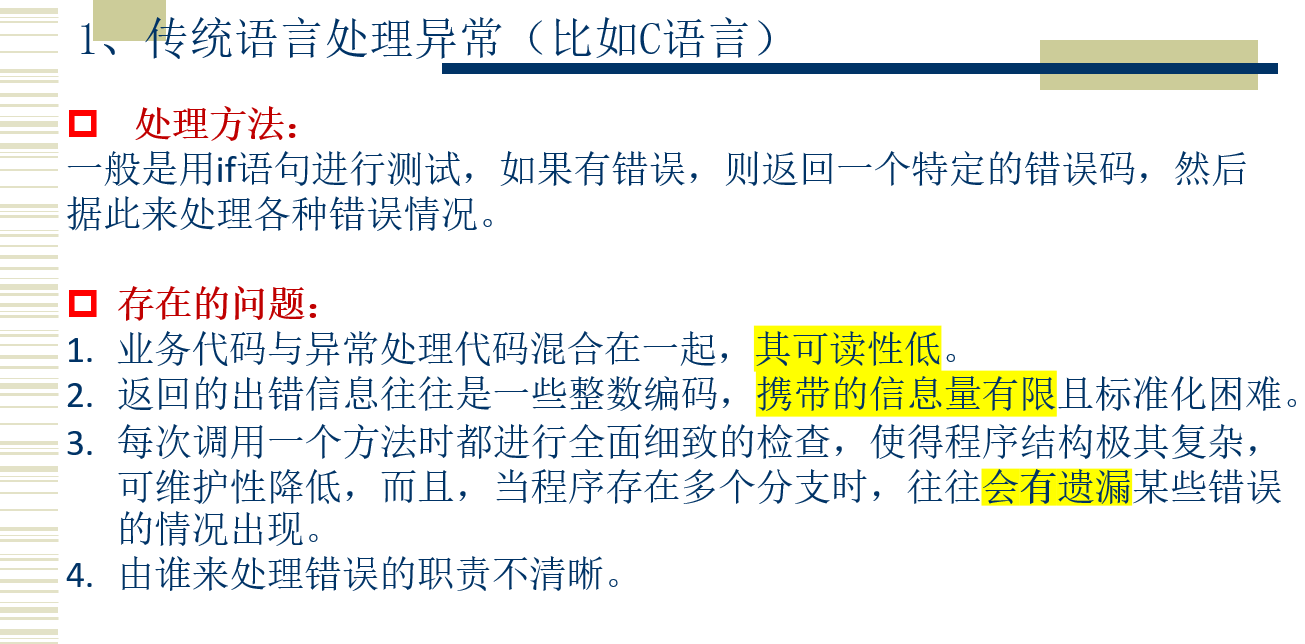




# 第五章 异常处理

1.哪些异常

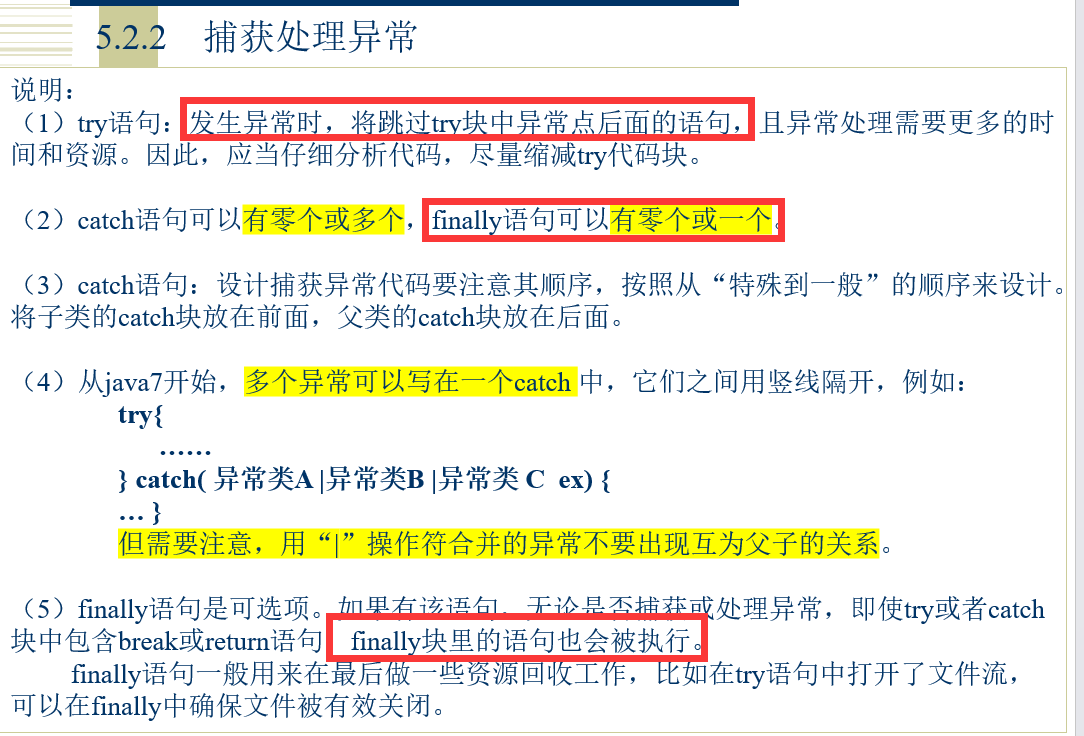
2.为什么采用这种异常处理机制

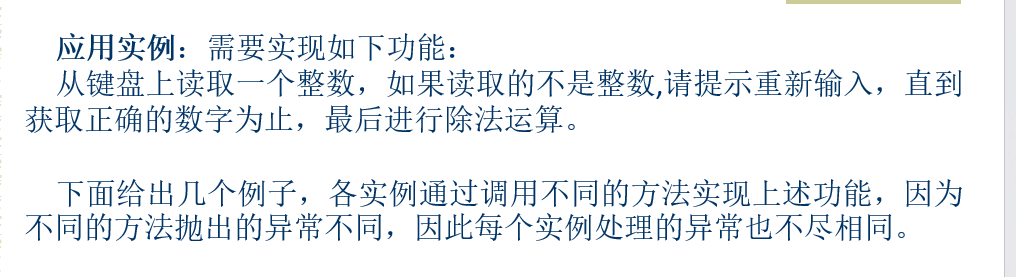


3.异常处理的方法与模板

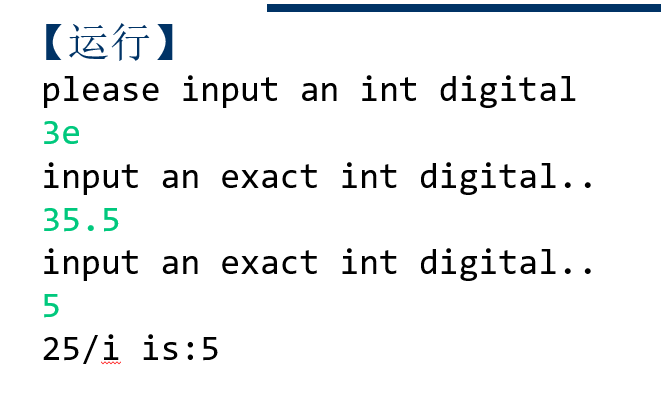


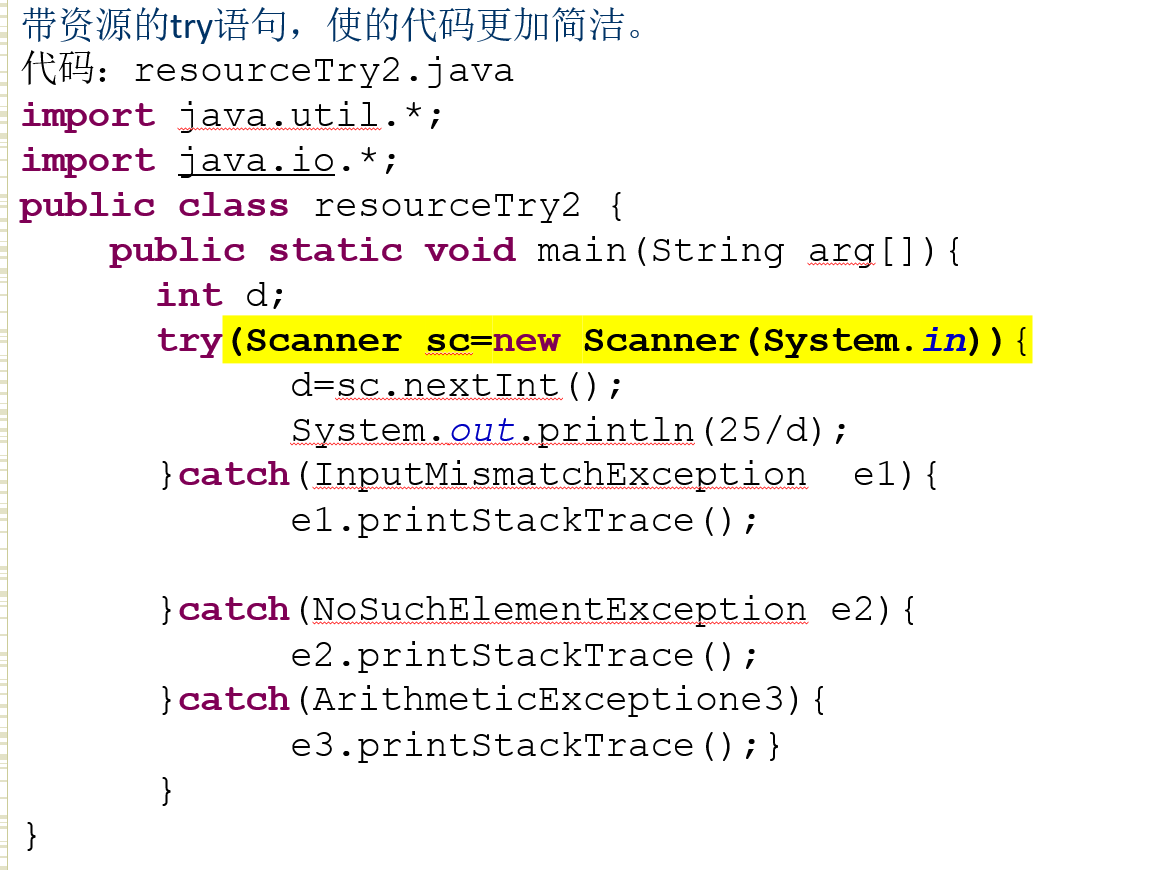


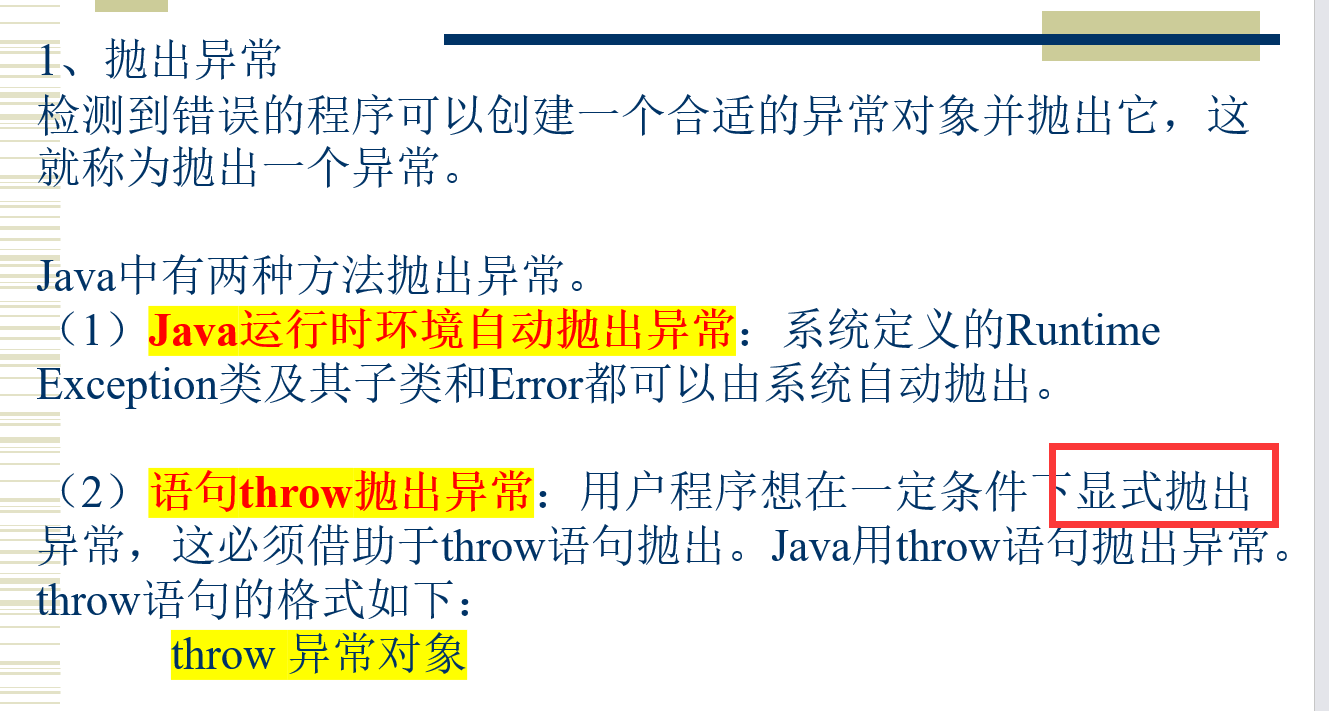




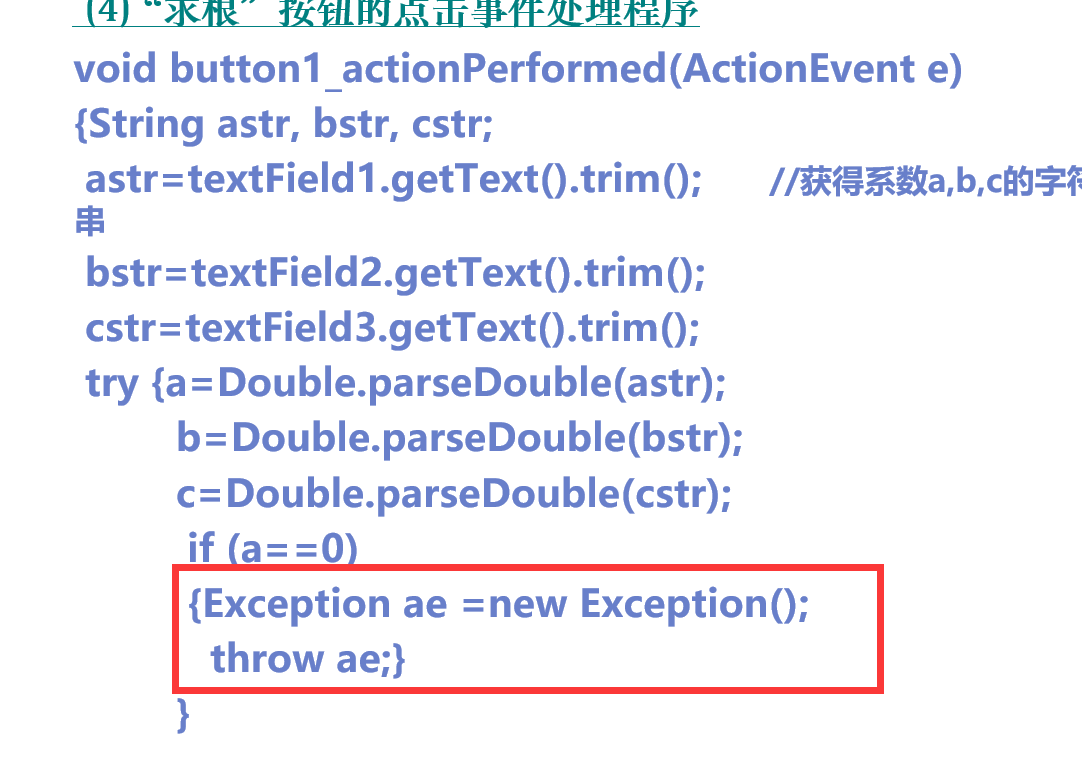




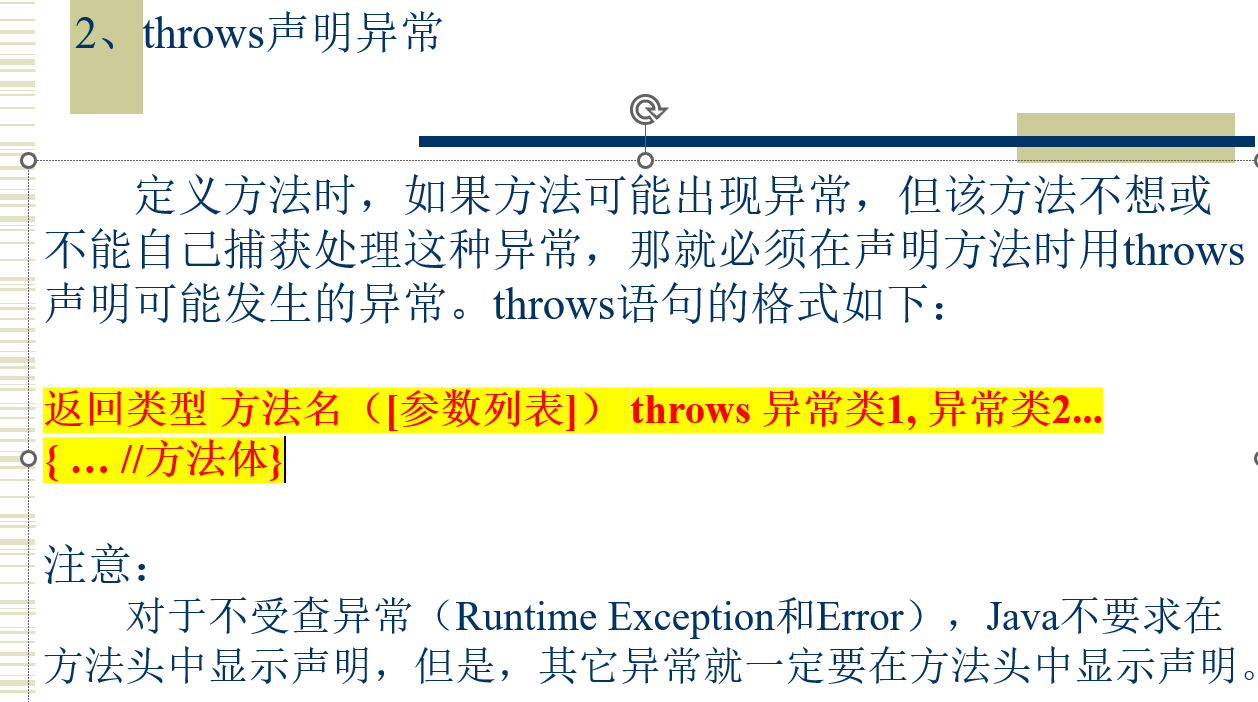




显示抛出异常







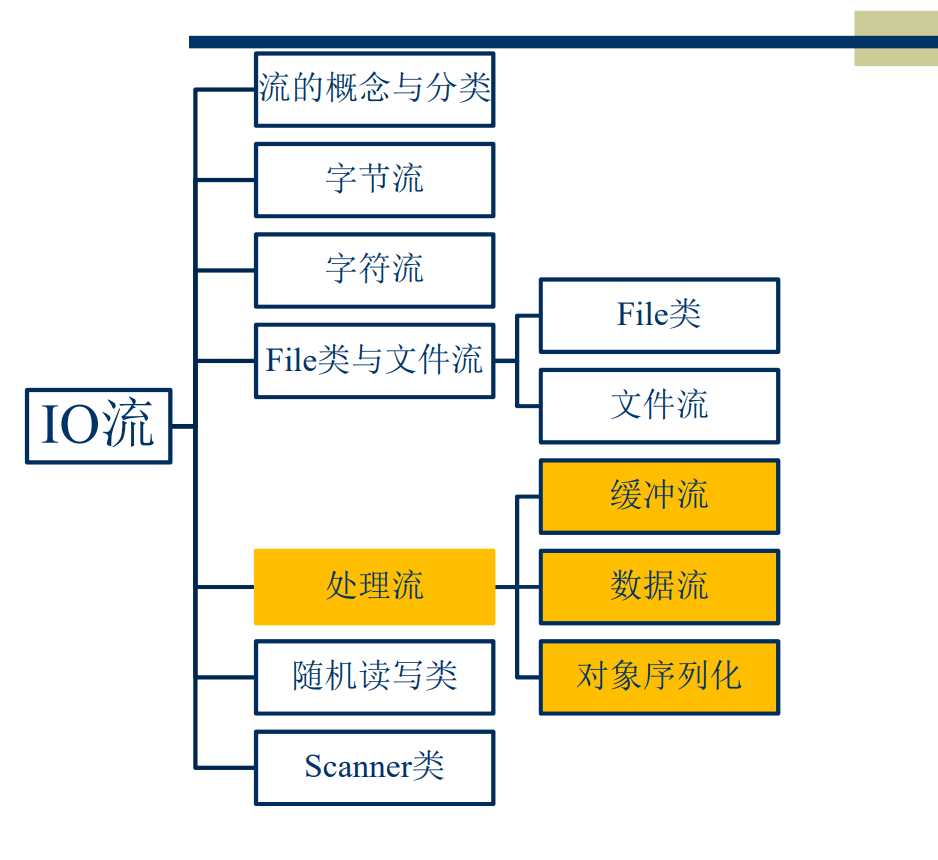
# 第六章 常用类



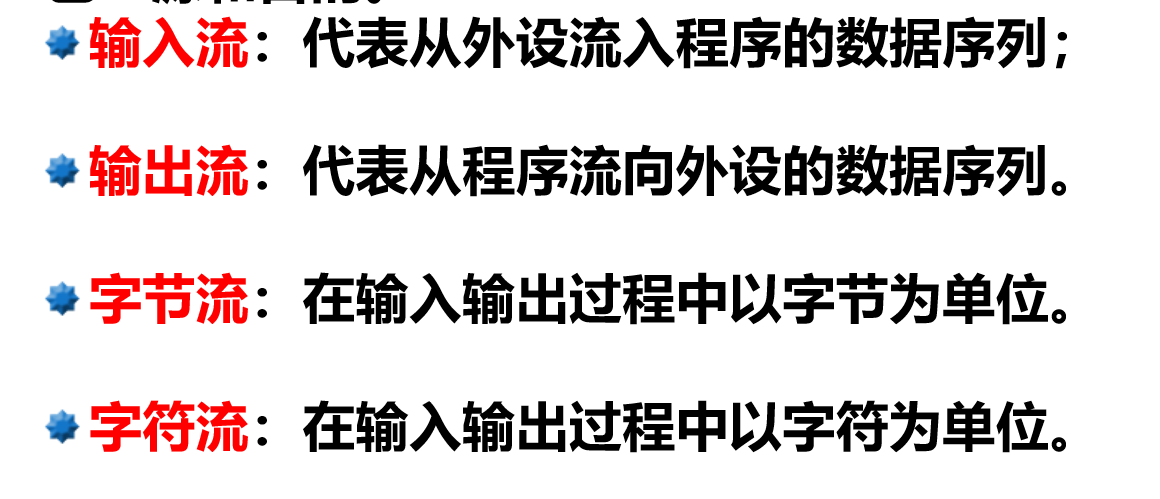


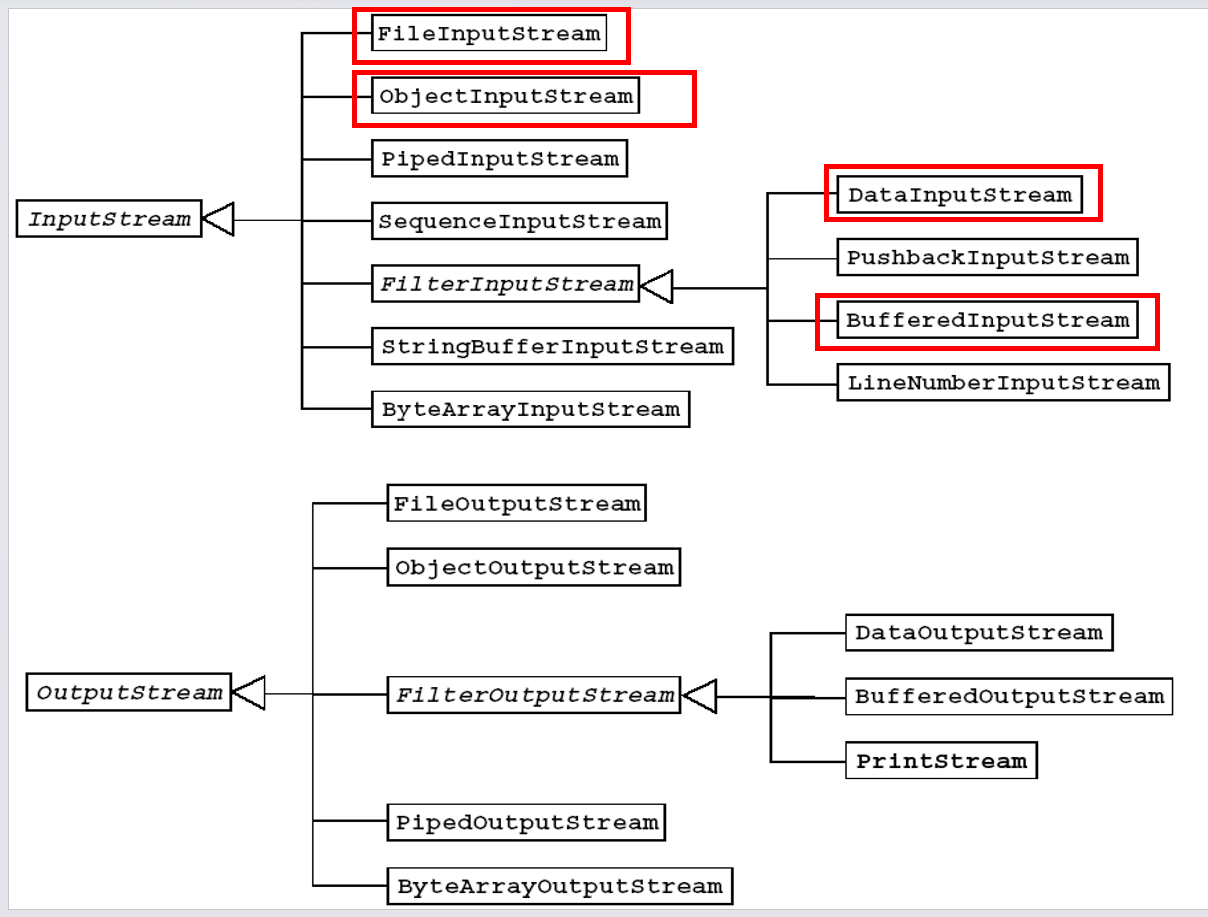
# 第七章 IO

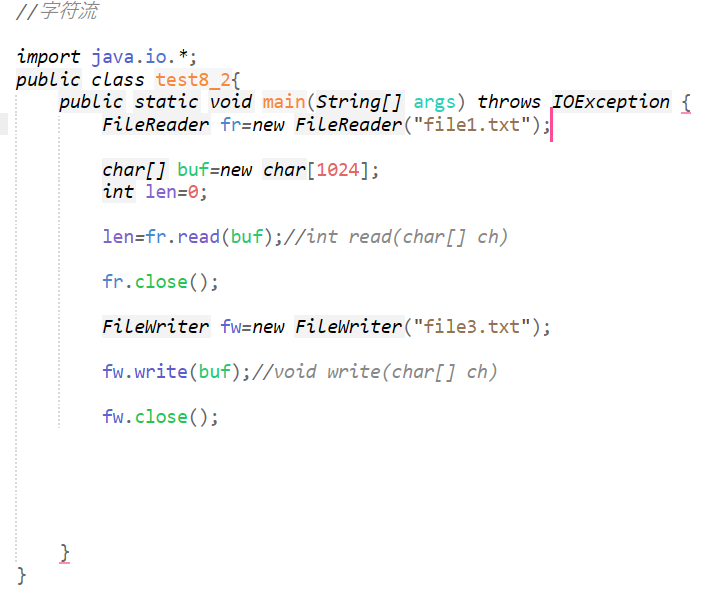
1.流的分类，特性



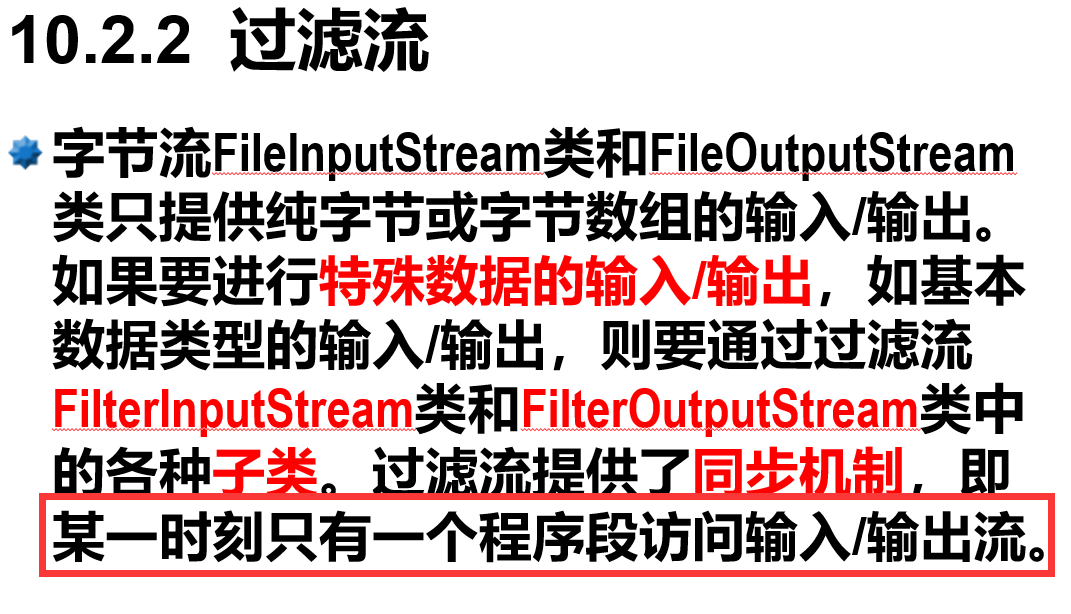


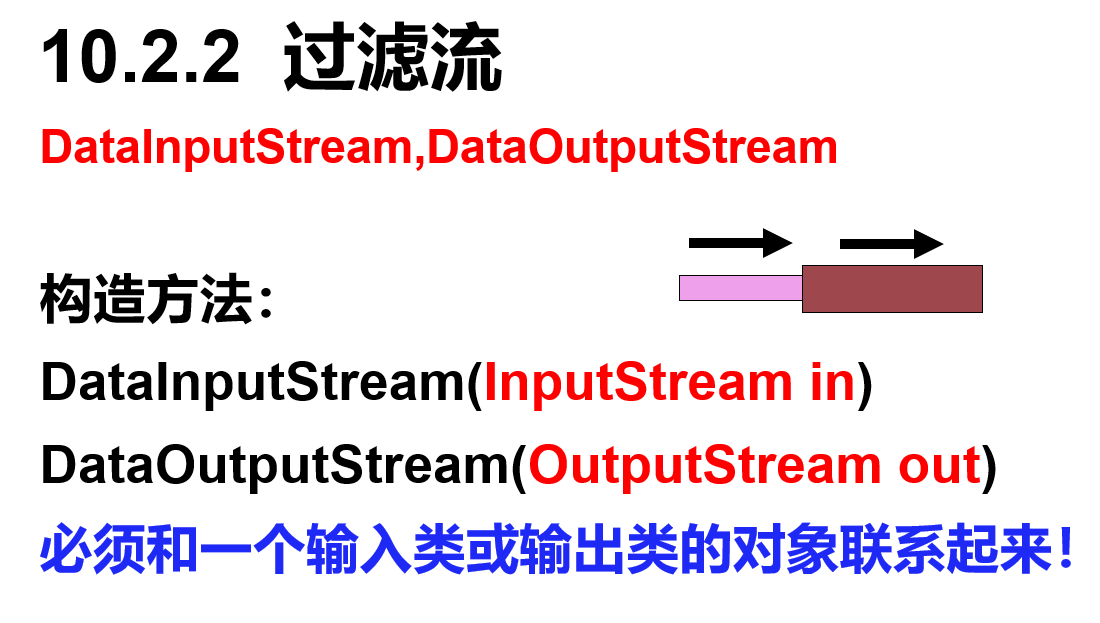


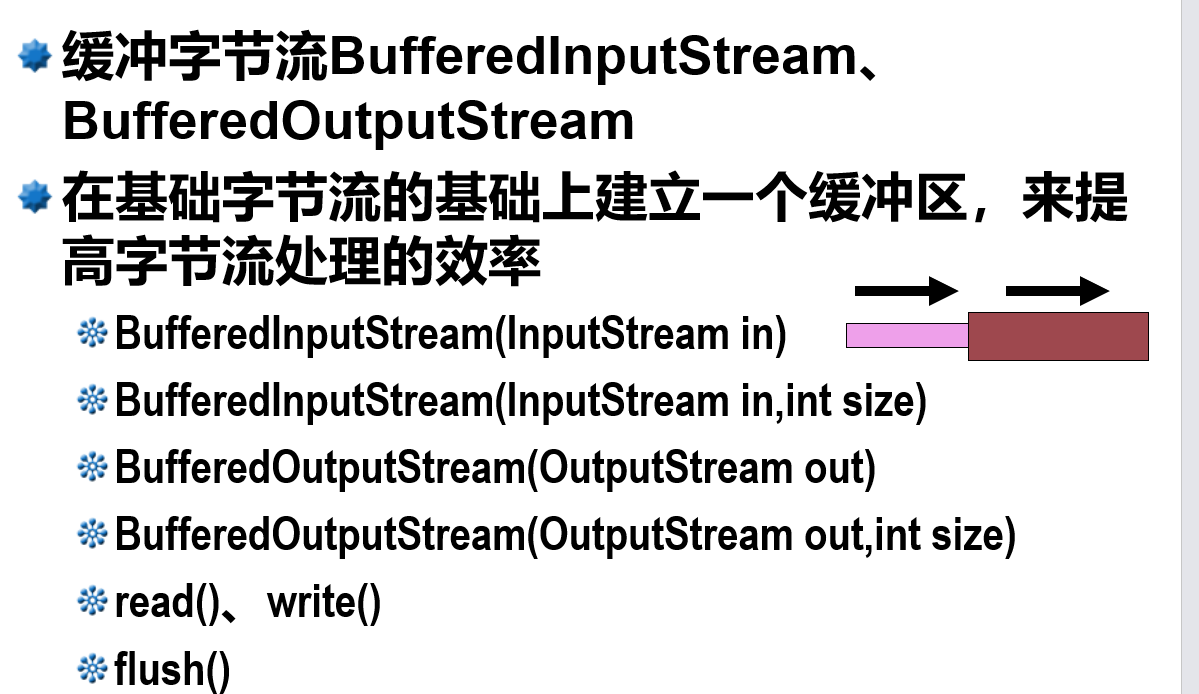


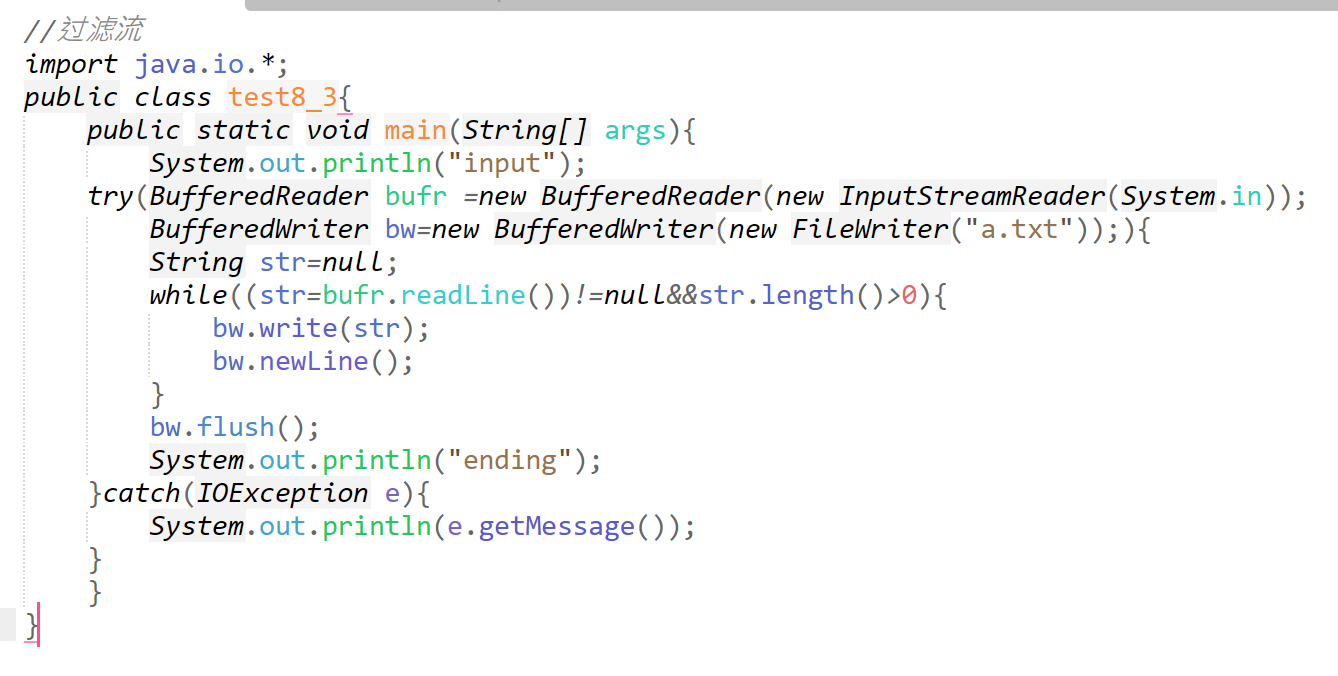


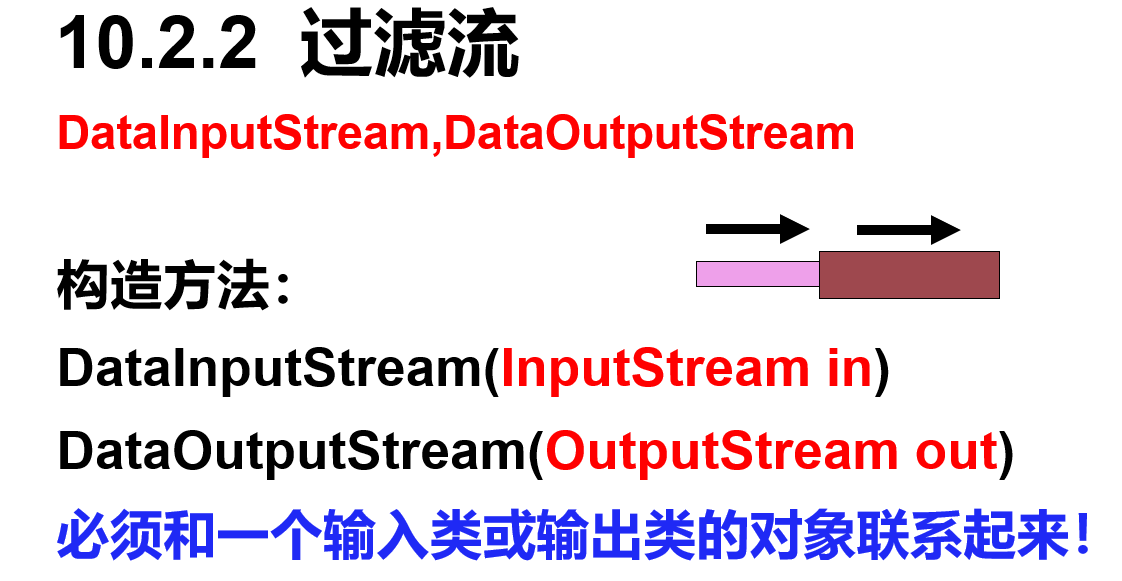


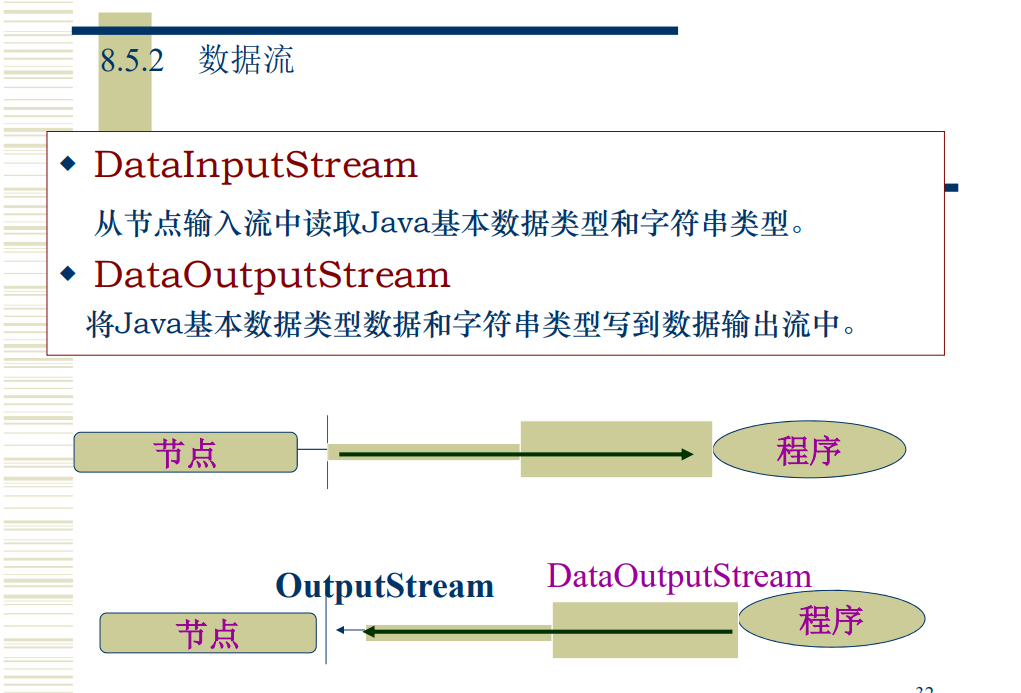








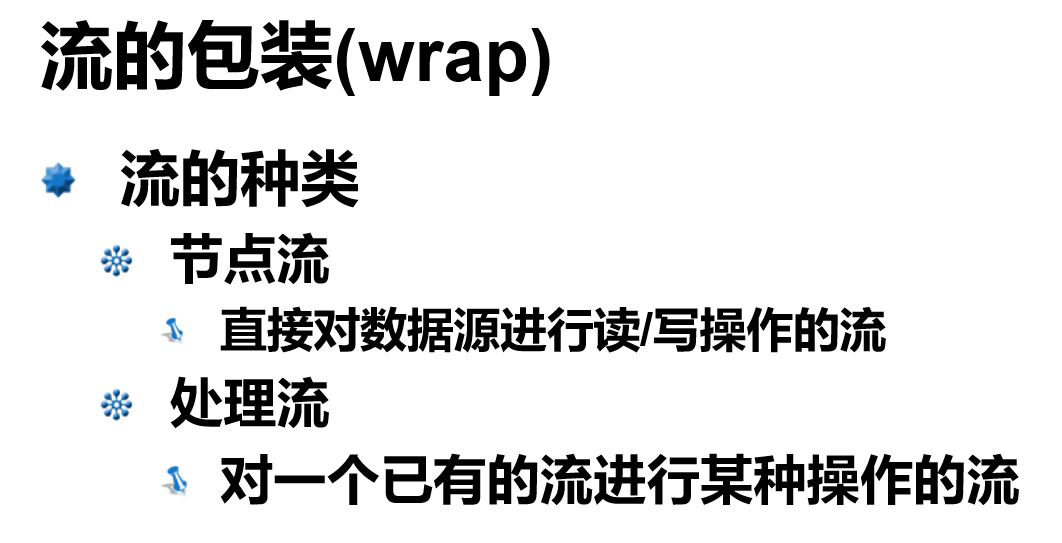


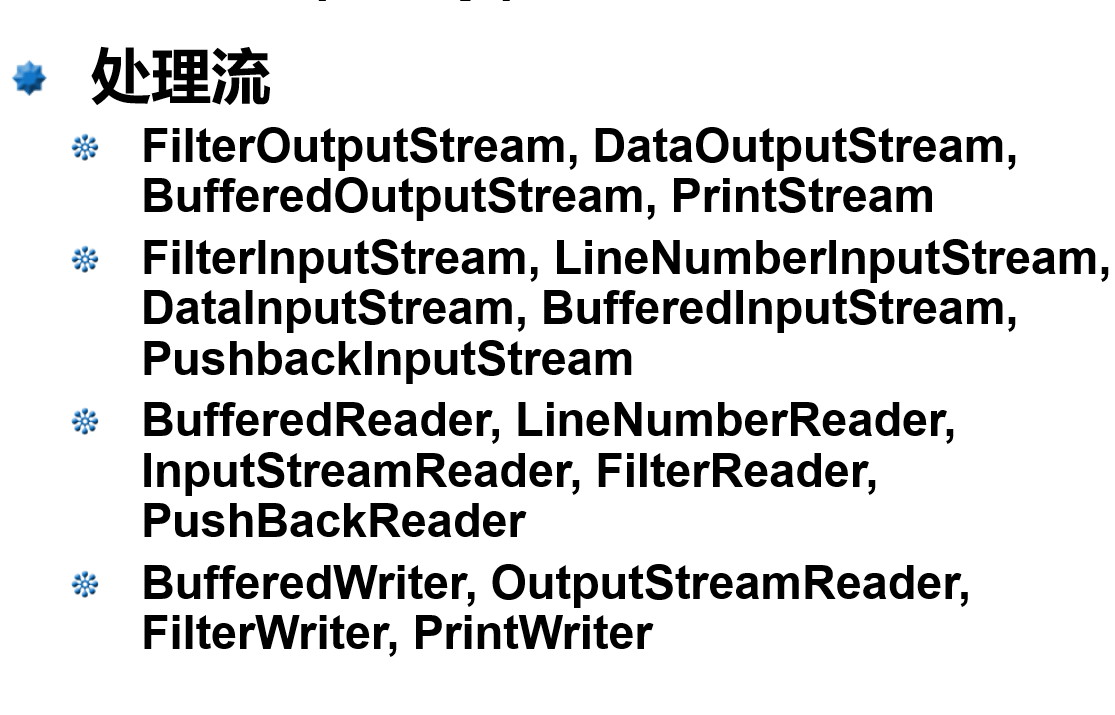




2.流的包装

包装的方法，尤其构造函数参数要对应，中间需要转换流（正确包装流）





流的包装，数据流的转换

# 第八章 GUI

1.控件（按钮，文本标签等）

JFrame JButton(基于swing的图形gui组件)，Label,textfield

2.布局管理器

Jframe默认BorderLayout

了解BorderLayout和FlowLayout

学会用这个布置基本界面，基本原理，实现位置的摆放，怎么更改

3.事件处理

委托机制的处理要充分了解原理

# 第九章 多线程

