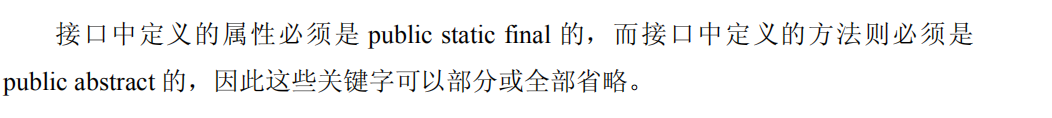
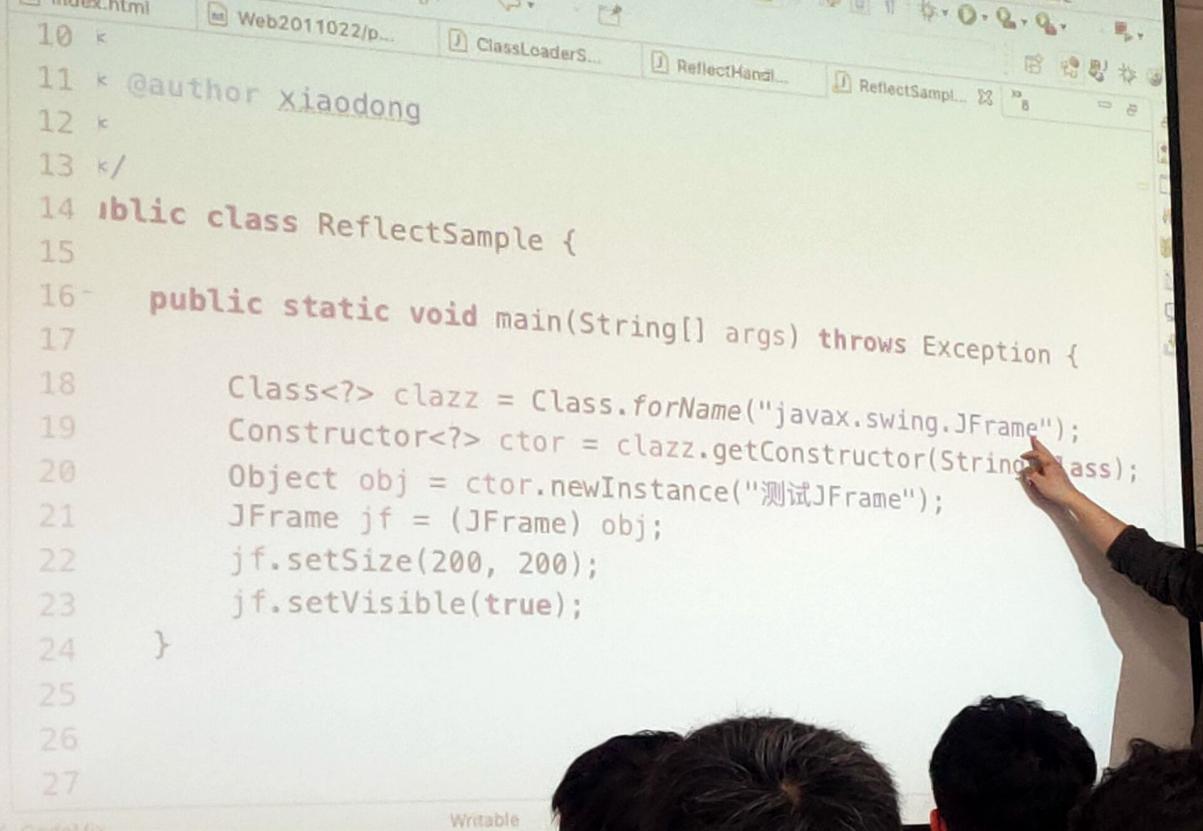


1. 填空题 :  18  
   [  
     "在重写Comparable接口的compareTo()方法定制比较逻辑时，需要确保其等价性判断与方法（ equals()  ）保持一致。",  
     "位于不同的包内，只有（ 父类声明为public时子类   ）的对象才可以直接访问其父类的protected成员。",  
     "引用类型数据值之间的强制类型转换称为（ 造型   ）。",  
     "toString()是在（ Object   ）类中定义的方法？",  
     "如果子类的构造方法中既没有显式地调用父类构造方法，也没有使用this关键字调用同一个类的其他重载构造方法，则系统会默认调用（ 父类无参构造方法   ）。",  
     "列举AWT中GUI组件的两类布局管理器（ BorderLayout   ）和（ FlowLayout   ）。",  
     "符合单例设计模式的类，其构造方法的访问控制权限要设定为（ private  ）的。",  
     "系统根据运行时对象的真正类型来确定具体调用哪一个方法，这一机制被称为（ 虚方法调用   ）。",  
     "String类是通过（ char数组 ）来保存字符串的。对于采用+运算符连接字符串的操作，编译器会主动使用（ StringBuilder  ）类进行优化，以提升字符串连接的性能。",  
     "命令行参数将被系统接收并静态初始化为一个（一维的String数组   ）对象，然后将之作为实参传给应用程序入口方法。",  
     "包含抽象方法的类必须声明为抽象类，用关键字（abstract  ）修饰。",  
     "ArrayList、Vector、HashMap、Hashtable中，不是线程安全的容器是（ArrayList，HashMap  ）。",  
     "在类的定义体中，方法的外部可包含static语句块，其执行特点是（ 当静态代码块所在类的字节码被加载进JVM时，就立马执行静态代码块，而且只会执行一次。   ）。",  
     "Java类之间的关系分为依赖关系、（ 聚合 ）和（ 泛化 ）。",  
     "可以使用Thread类的（ join（） ）方法实现线程的串行化，使用（ yield() ）方法实现线程的让步。",  
     "（  Throwable   ）是Exception和Error类的父类，它提供（ printStackTrace() ）方法跟踪异常事件发生时运行栈的内容，并将相关信息输出到标准错误输出设备。",  
     "JVM栈内存可以存放定义的基本类型的变量和（ 引用类型的 ）变量。",  
     "泛型机制其实质是将原本确定不变的数据类型（ 参数化 ）。"  
   ]  
   试题题型 :  1  
   [  
     "Java期末考试试卷题型包括：填空题、判断题、问答题和手写代码题

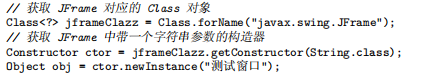
多线程编程，文件（流），filter，。"  
]  
简答题 :  1  
[  
  "Java提供的两种基本的创建线程方式是什么？"  
]



SM16WOJYX2`3D9LVJ`090R5



↑方法名是干什么的



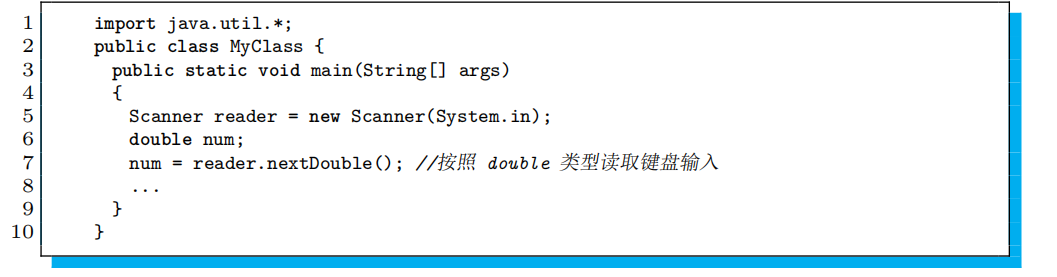
构造一个class类型对象 ，<?> 叫泛型 ， ？代表一个通配符

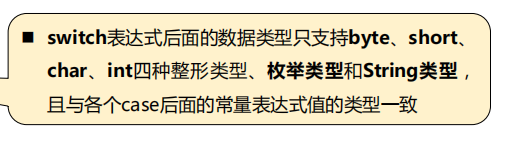
getConstructor获取构造器 有一个字符串参数的构造器

用这个构造器创造一个实例

123行相当于 new JFrame("")

区别在于，用这种方法可以在运行时确定



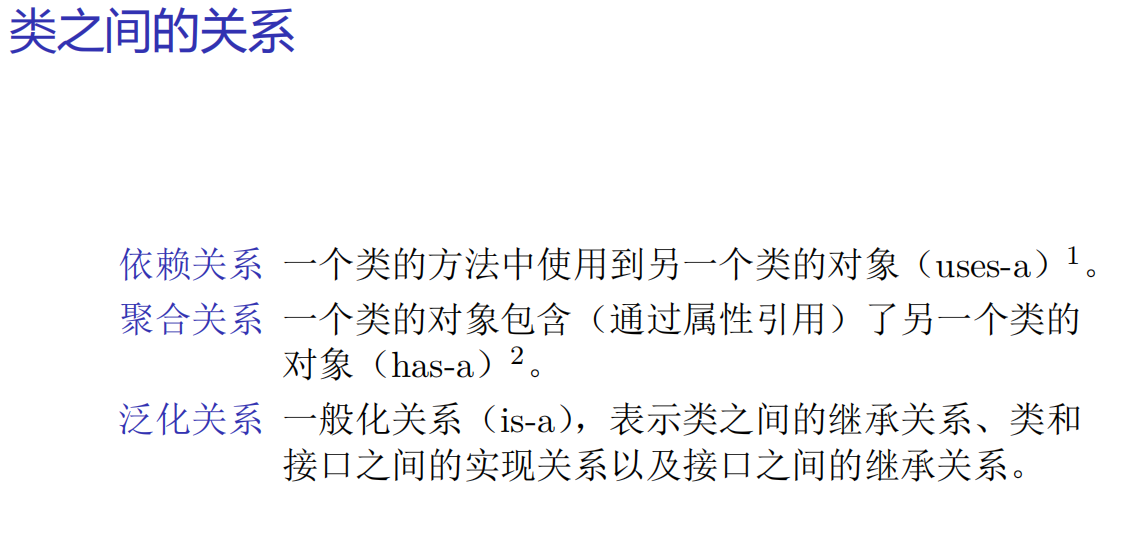




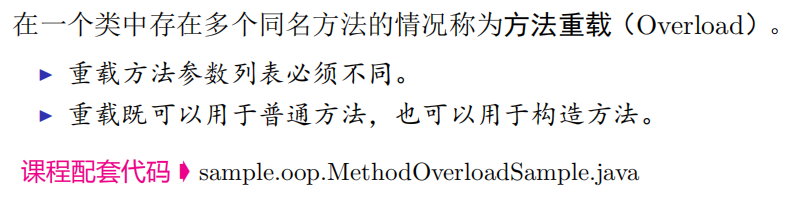


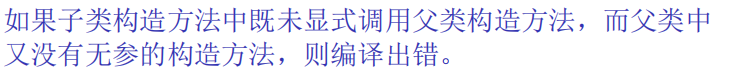


类之间的关系



方法重载



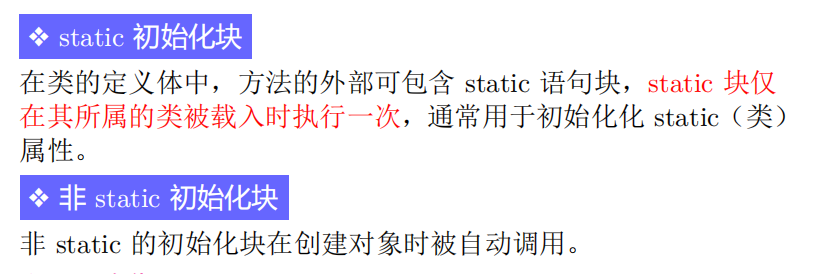


重载返回类型可以不同

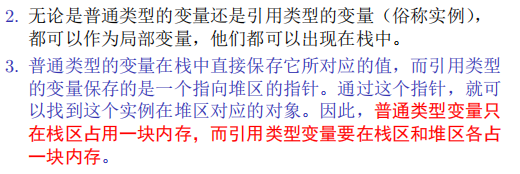
不允许参数个数或参数类型完全相同而只有返回值类型不同的重载

static属性和方法

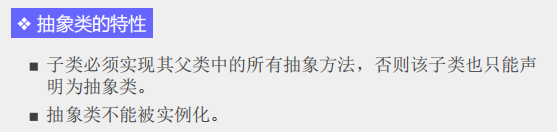


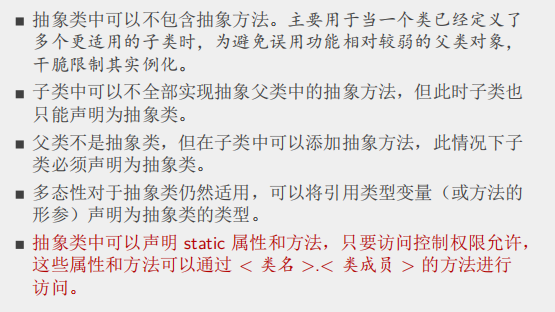


java基本数据类型和引用类型的内存占用

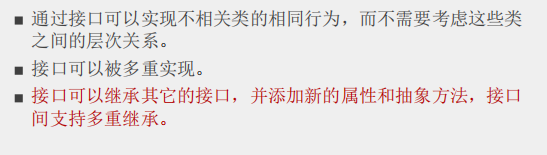


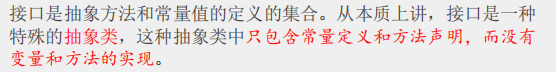
抽象类特征和接口特征





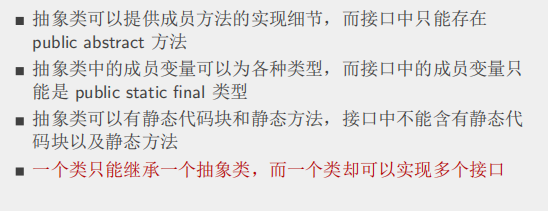
抽象类可以有构造方法，但不能创建实例



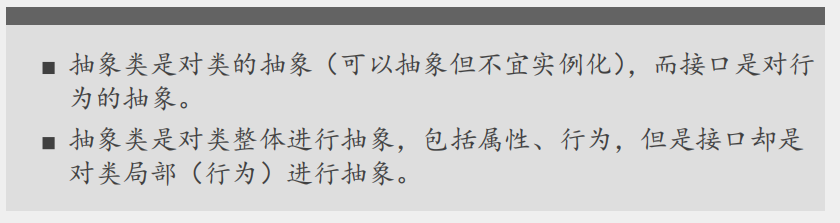


接口和抽象类的异同

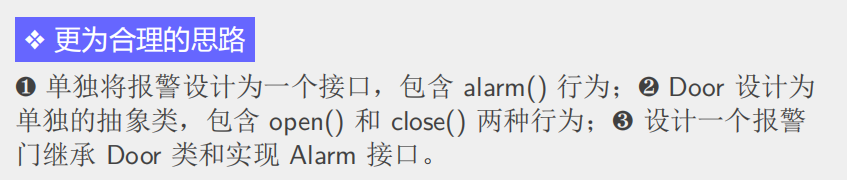
1. 语法区别



2.设计层面



3.实例



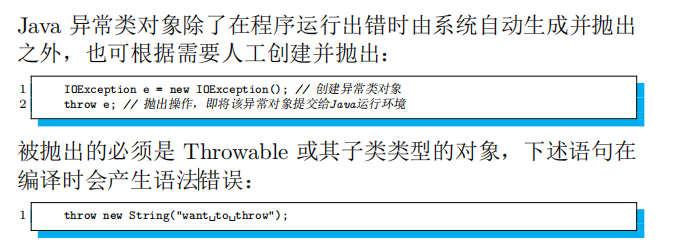
抽象类和接口的作用

提供模板提高开发效率

提高代码复用性

开发更加规范

人工抛出异常



IOException e = new IOException();

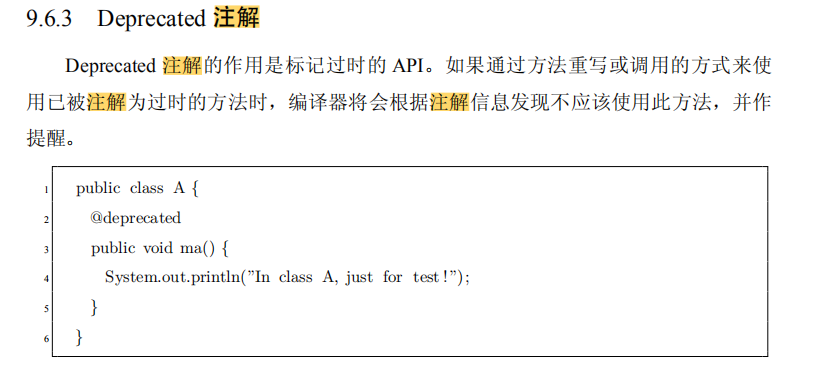
throw e;

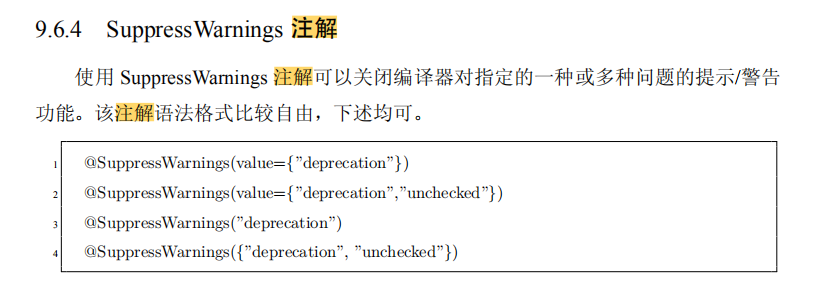
Java 系统属性对象：Properties，如何取得该对象？

System.getProperties()

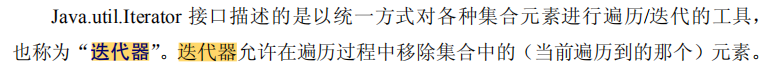
java常见注解含义



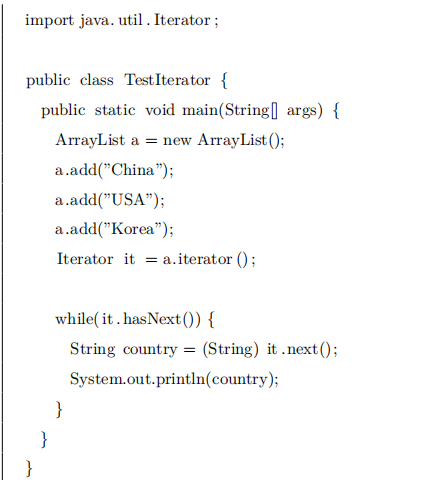




迭代器





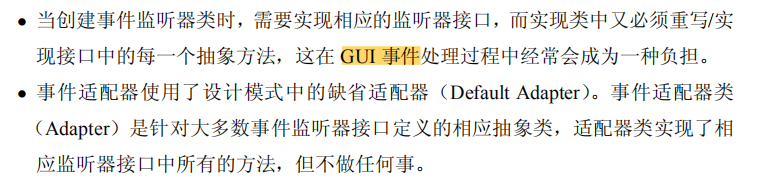


JAVA GUI编程可用类库

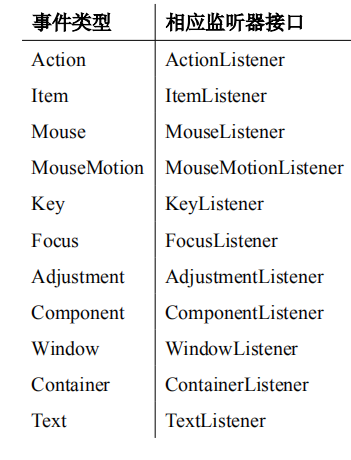
java.awt.event

javax.swing

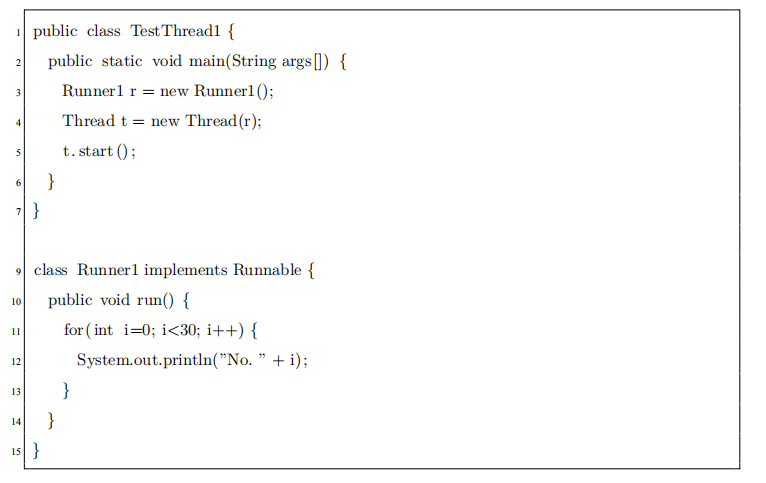
GUI事件适配器存在意义

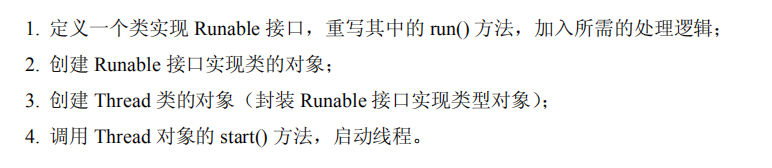


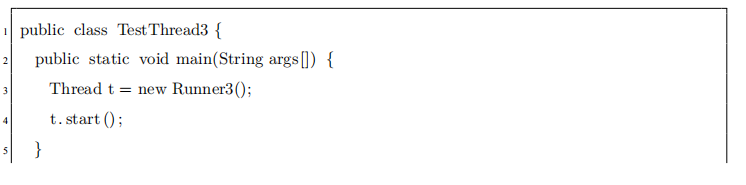
GUI事件常见监听器

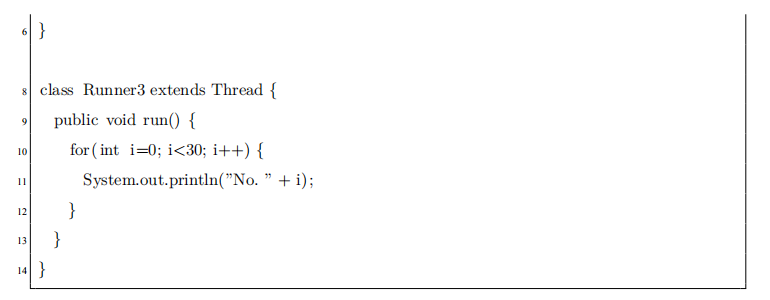


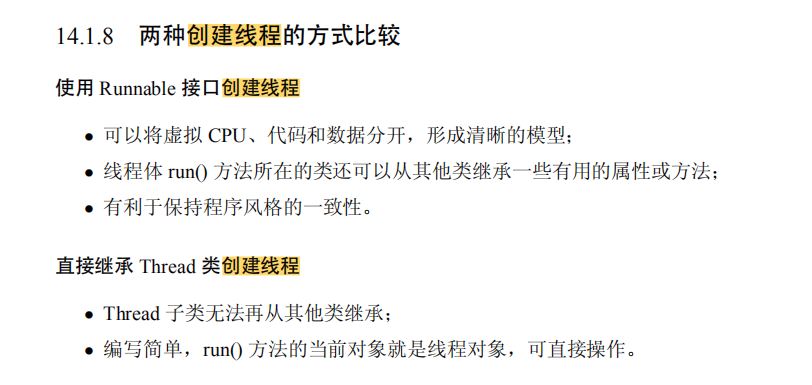
线程实现两种方式：

1. 

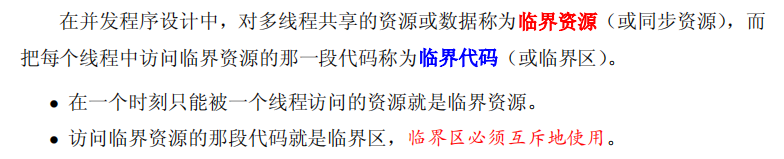


1. 

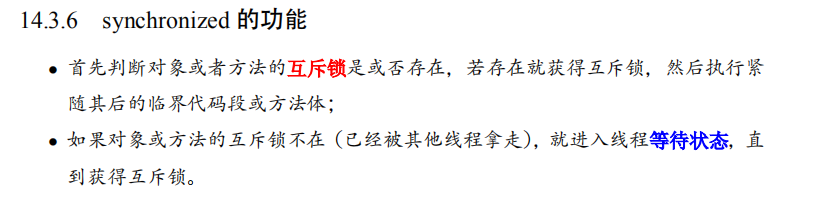


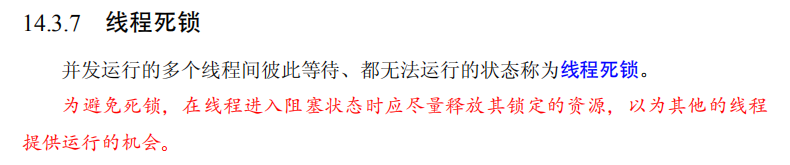


关键字synchronized，临界资源，锁和死锁

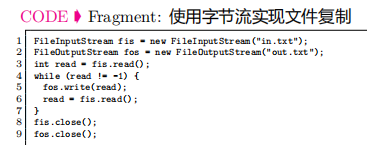


包括对象和方法



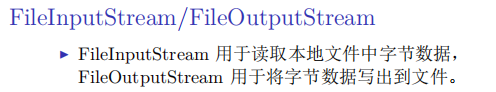


线程编程：文件复制



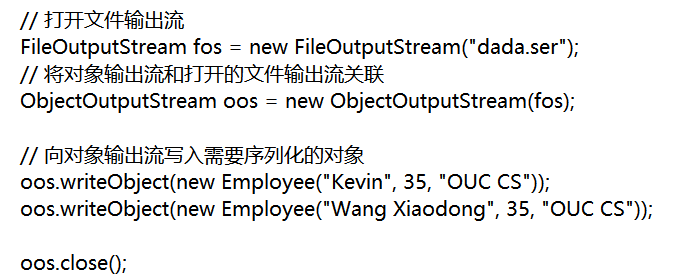
常用IO流类型

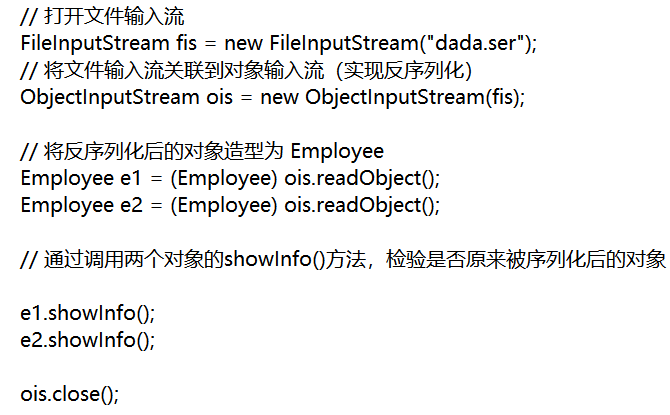




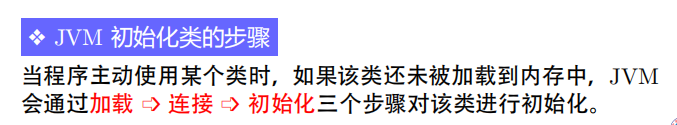
对象序列化概念及编程实现







获得class对象的方式



Class.forName()

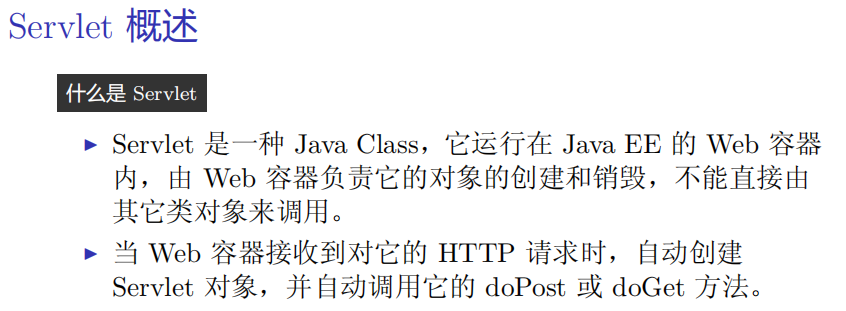
Person.calss()

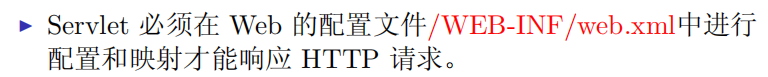
p.getClass()

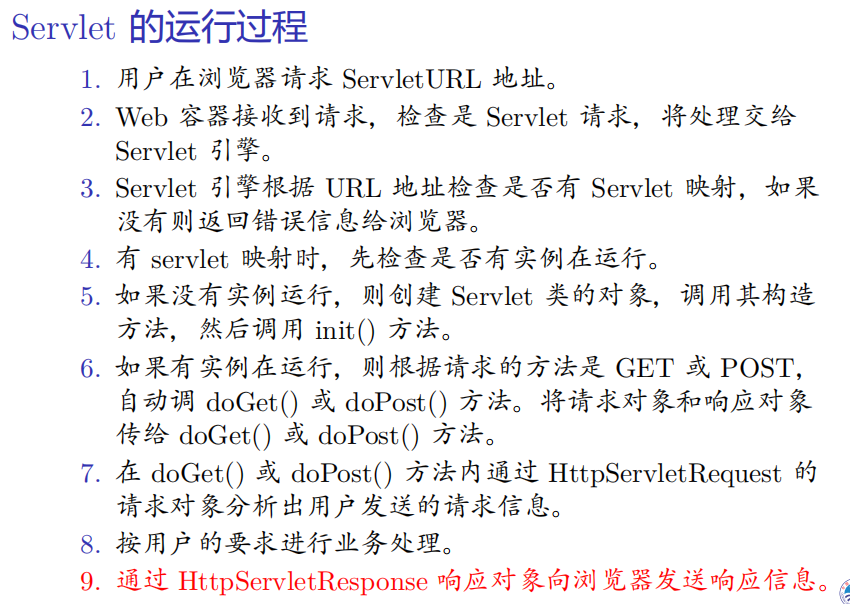
反射编程：见代码解释

JAVAEE

Servlet概念和工作原理







Servlet编程

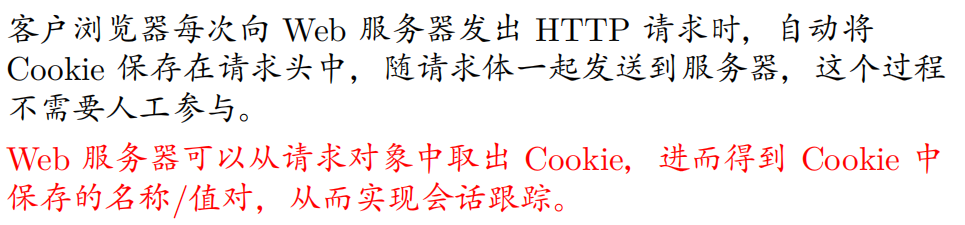




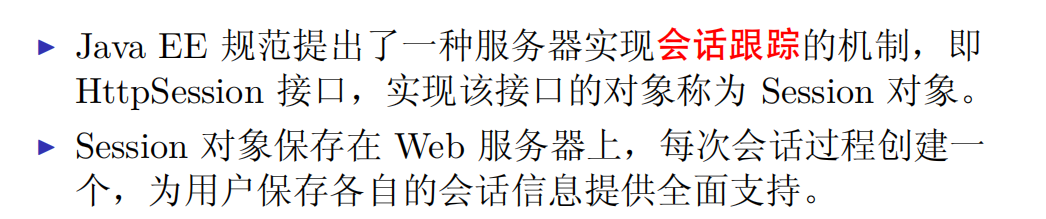
JAVA EE会话跟踪方法

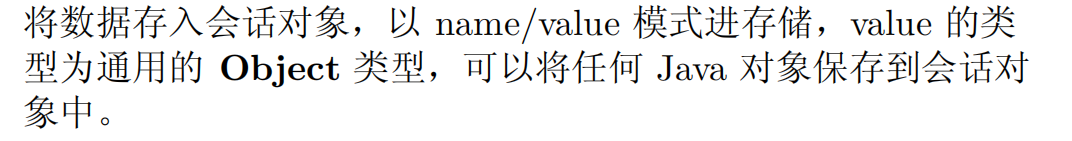
URL重写

cookie



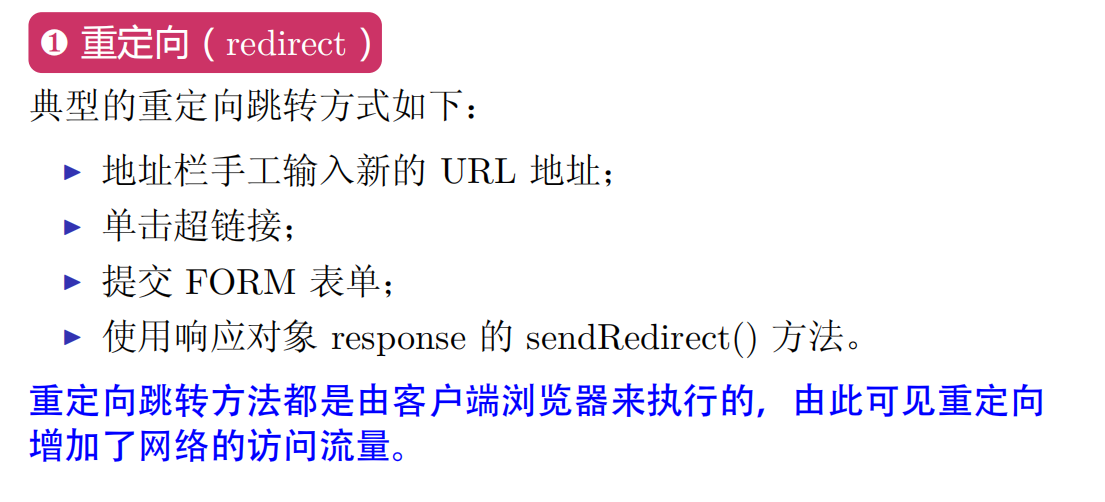
javaEE会话对象

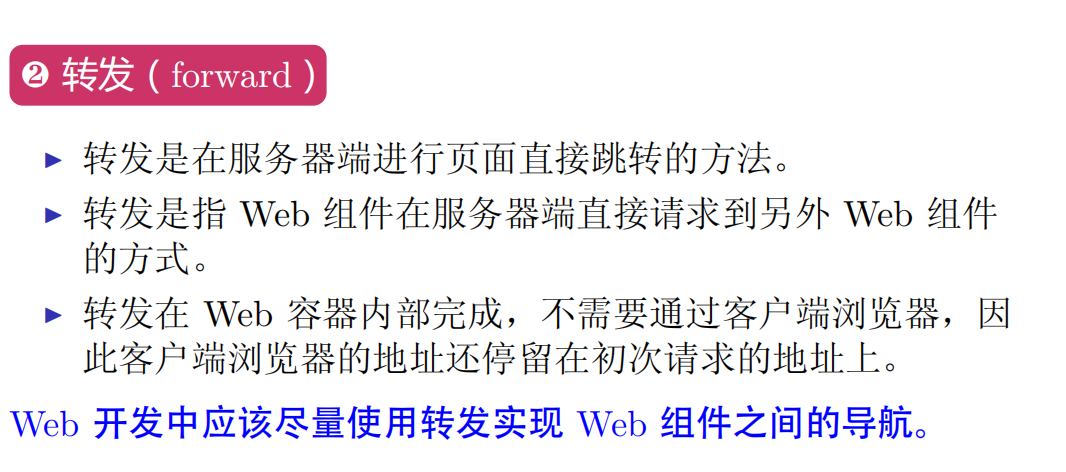




解决：http协议本身无状态性

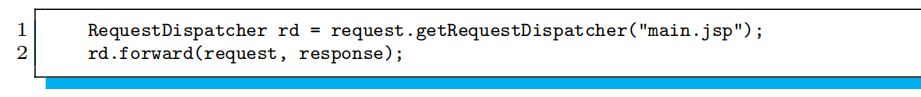
java web 跳转的两种方式：重定向和转发



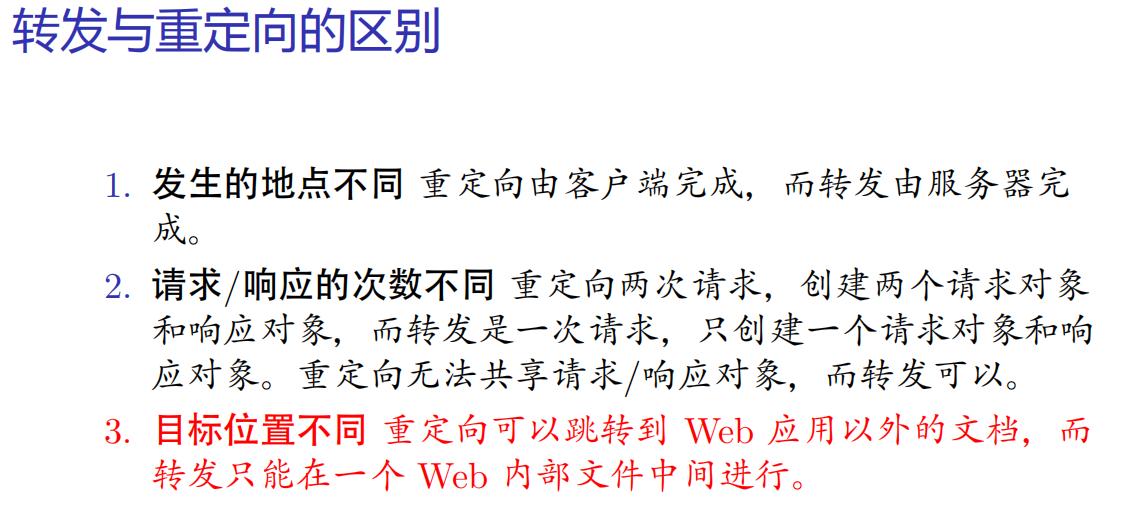


取得转发对象

forward转发



转发与重定向不同



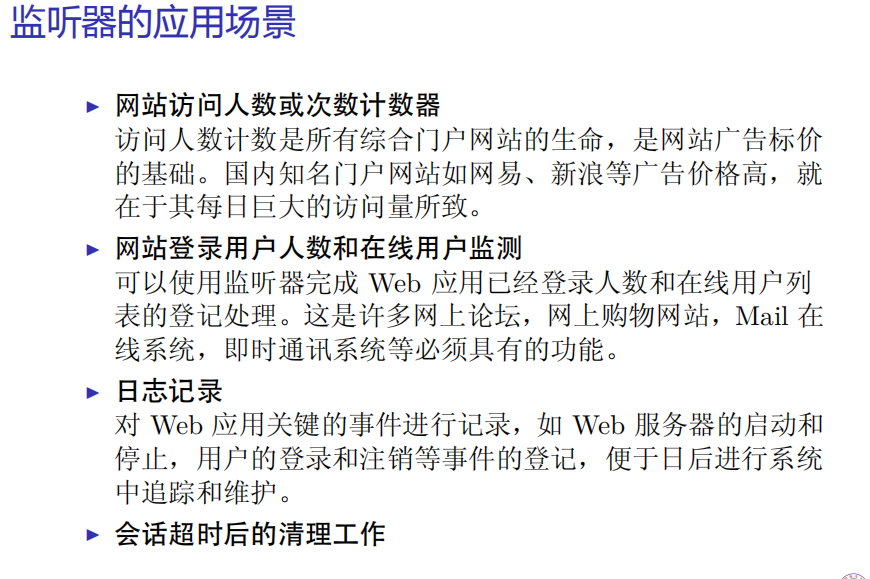
Servlet之间的数据共享



过滤器编程

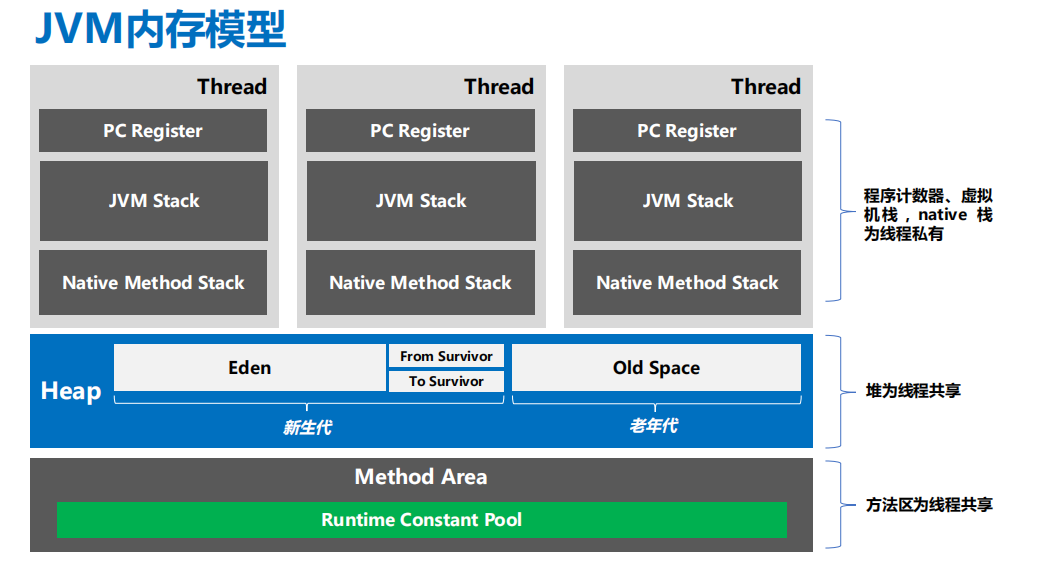


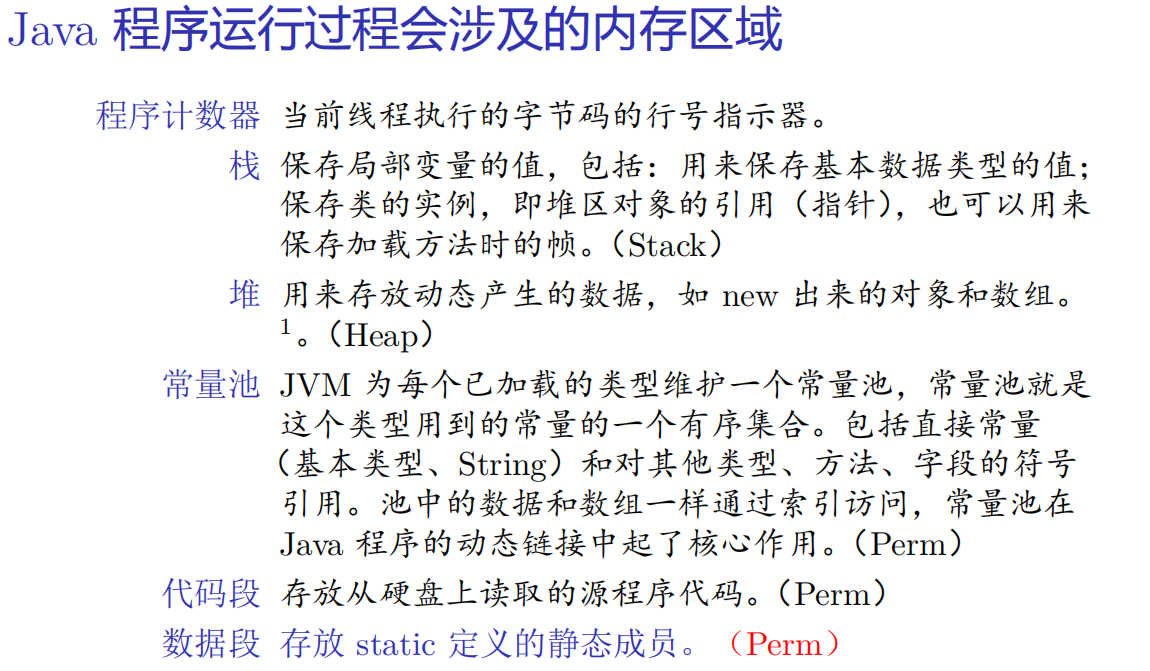
监听器应用场景，列举目前Servlet API提供的监听器





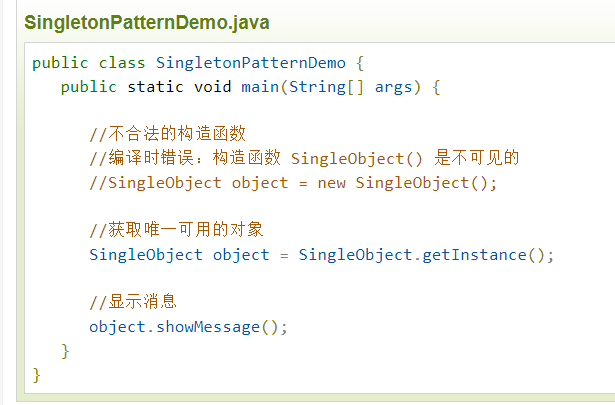
JVM内存





单例模式





==和equals的区别

MVC

选择题10道

填空10道

简答 考核心概念 一道5分 一定要写东西

代码（反射，过滤器） 代码解释

反射

涉及过滤器的代码，方法的作用 doFilter以及里面的doFilter

关键api

设计模式：单例模式

需要总结的内容：

1. 线程 创建线程的方式方法：有哪些方法，可以从什么转态转移到什么状态
2. 与锁有关的 Object提供的几个方法 wait notify notifyAll
3. 内存 JVM内存区域有哪些 都是干什么的 哪一些线程共享/独立
4. 堆内存里小的结构
5. MVC
6. 