# 面向对象:

### 1.1

对象的产生：在所要解决的问题身上提出所有概念性的组件，在程序中将其表达为一个对象。

程序是一大堆对象的组合。

### 1.2对象的接口

我要建立一个什么样的系统，根据具体系统抽象多个类，建好类后根据情况延伸出许多对象

但是怎样通过对象来做一些实际的事情。？？？？？？？？？？？

类型和接口是等价或对应的关系。

### 1.3实现方案的隐藏

控制对成员的访问原因：首先隐藏一些属性信息，用户只需要操作接口即可，其次允许后期修改库中类的内部结构，却不用担心对客户程序员（调用该库的人）的影响（前提接口与实现方法早已隔离，并分别受到保护）。

Java三个显式关键字来 private public protected

Protect同一个包中或者继承类可以访问。

## 1.4方案的重复使用

在新类中添加一个“成员对象”，一般进行private，他比继承要更加灵活。

### 1.5继承：重新使用接口

在创建一种类型的情况下，需要在创建一种实现大致相同功能的类型时可通过继承来实现。子类包括了父类的公开成员（private除外），复制了父类的接口。

通过继承产生的一系列类都建立在具有统一接口基础上。

### 1.6多行对象的互换使用

**即多态的体现**







**动态绑定**

即为多态自动识别传入的参数类型，并调用相应的方法。







最后一句会调用c的方法。

**抽象的基础类和方法**

抽象使用abstract关键字，对类声明为抽象类，不在允许创建一个对象，同时也可对方法进行描述，声明为一个尚未实现的方法。

Interface关键字将抽象类的概念进一步进行了延伸，完全禁止了所有函数的定义。

### 1.7对象的创建和存在时间

### 1.9多线程

指同时对多个任务加以控制。

拥有极其底层知识的程序员，通过编写中断服务来进行任务控制。中断对于实时性很强的任务是有必要的。

将实时性的任务划分为独立运行的程序片段，该片段即为线程。

线程的难点在于：共享资源 引入锁的概念。

### 1.10永久性

对象的序列化和反序列化

### 1.11 java和因特网

Java解决了传统的程序设计问题，同时解决了web编程问题。

客户机/服务器

客户端编程（针对浏览器）：

1、浏览器插件：通过插件位浏览器添加新的功能。

2、脚本编制语言：通过脚本语言，可将客户端程序的源码直接插入到html页，对语言解释的插件会在html显示的时候自动激活。

脚本语言倾向于特定类型问题的解决，比如JavaScript：

服务器编程：

Java进行的服务器端编程，写出的程序叫做servlet.

### 1.12分析和设计

项目中时刻提醒着自己对象是什么（怎样将自己的项目分割成一系列兰度的组件）？

他们的接口是什么（需要讲什么消息发给每个对象）？

阶段0：拟定一个计划，在整个过程中设置几个标志。

阶段1：确定制作的核心任务。





时间的估计：感觉需要的时间加倍在加上10%。

阶段2：使用“统一建模语言（UML）”，比如处理一些图表，流程图。（做出对对象以及他们的接口的说明）