

面向对象编程 Java

Yuanbin Wu
cs@ecnu

面向对象编程 Java

- Java 简介
- 面向对象编程概述
- 课程介绍

Java ?





Kopi Jawa (Java Coffee)



编程语言



Java 简介

- 1995 年, Sun 公司发布 Java 1.0
 - James Gosling, Mike Sheridan, and Patrick Naughton
- 2010 年, Oracle 收购 Sun
- 目前版本: Java 8

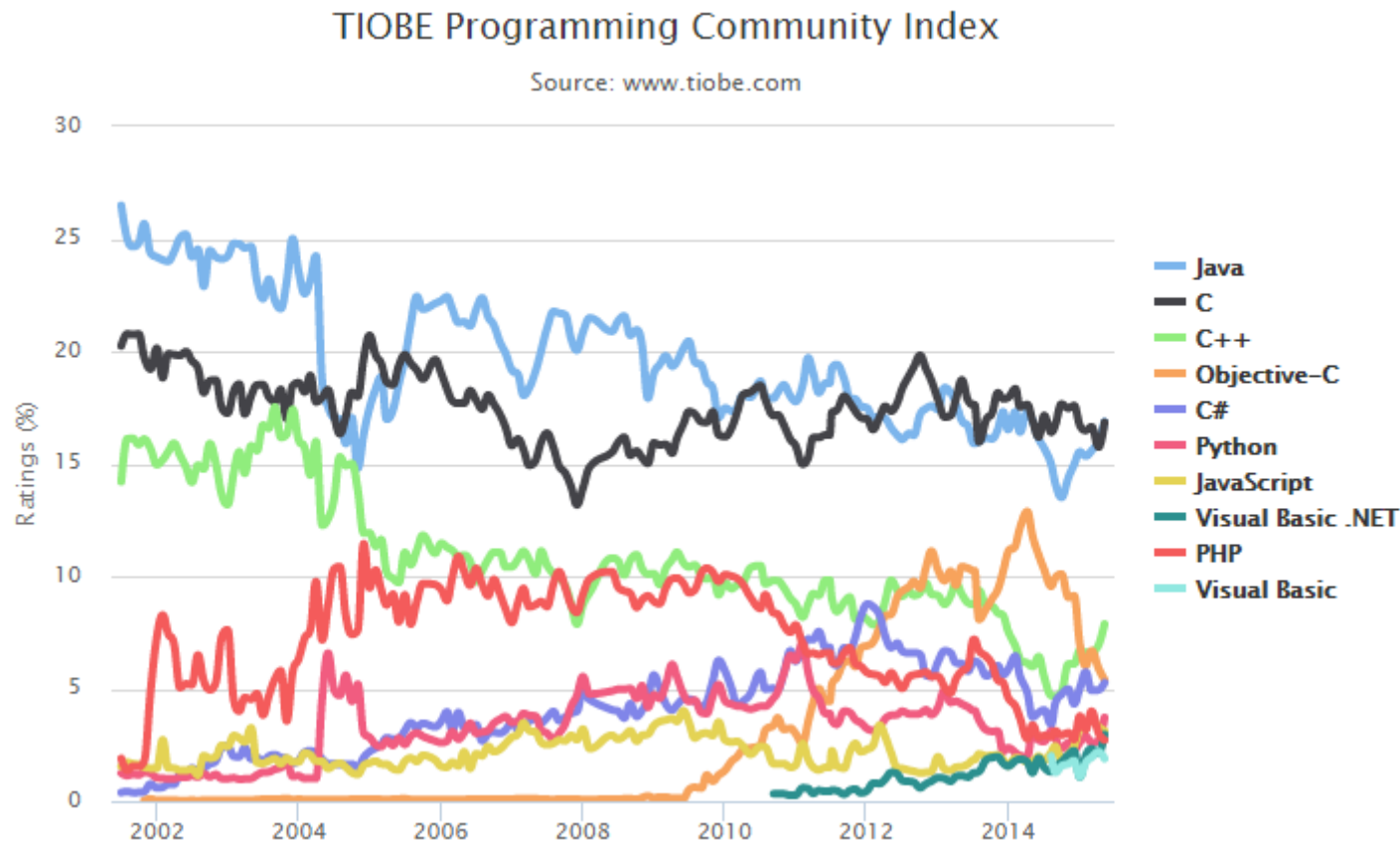


Java 简介

- 特点
 - 面向对象
 - 跨平台
 - 类 C 语法
 - 自由软件

Java 简介

- 特点：面向对象 (Object-oriented, OO)
 - 支持面向对象编程语法
 - 使用广泛的面向对象语言之一



Java 简介

- 特点：跨平台
 - “Write once, run anywhere” (WORA)
 - Windows, Linux, OS-X,...
 - 编译后的 **Java** 程序可以在所有平台运行！

Java 简介

- 特点：类 C 语法
 - 变量类型，表达式，控制语句与 C 相似

```
int []a = {1,2,3,4,5};  
int []b = {6,7,8,9,10};  
for (int i = 0; i < 5; ++i) {  
    if (i % 2 == 0) {  
        int t = a[i];  
        a[i] = b[i];  
        b[i] = t;  
    }  
}
```

- 减少底层功能
 - 指针
 - 更容易的内存管理

Java 简介

- 特点：自由软件 (Free Software)
 - 2007 年，Sun 让大部分 Java 技术使用 GPL 协议
 - 开源实现（例如：OpenJDK）

Java 简介

- 第一个 Java 程序
 - 输出 “Hello World!”
 - C 语言实现
 - 主函数 : main
 - 标准输出函数 : printf
 - 字符串 : “Hello World!”
 - 流程 :
 - 输入源码
 - 编译
 - 运行

Java 简介

- 第一个 Java 程序
 - Let's code
 - “类 C 语法”
 - Let's try it


Java 简介

- 编写 Java 程序的流程：
 - 编写源代码：文本编辑器
 - 编译：`javac`
 - 运行：`java`
- 可能出现的错误
 - 编译时错误 (Compile-time error)
 - 运行时错误 (Run-time error)
 - 逻辑错误 (Logic error)

面向对象编程 Java

- Java 简介
- 面向对象编程概述
- 课程介绍

面向对象编程概述

- 实际问题  计算模型
 - 实际问题 (Problem space)
 - 计算模型 (Solution space)

面向对象编程概述

- 问题：上海的马路上有多少井盖？
 - 一个 C 程序
 - 给定一个文件，每行一个记录，包含两列：路名，井盖数
 - 定义变量 `int sum = 0;`
 - 读入这个文件，每次一行，保存在字符数组 `c` 中
 - 在 `c` 中取出第二列，将其转换为整数 `s`
 - `sum = sum + s`
 - 循环直到文件结束
 - 输出 `sum`

面向对象编程概述

- 问题：上海的马路有多少井盖？
 - 一辆可识别井盖的车
 - 每条马路：是否已经走过？
 - 每个井盖：是否已被统计？
 - 多辆可识别井盖的车
 - 每条马路：是否已经走过？
 - 每个井盖：是否已被统计？
 - 每辆车：识别了多少井盖？

面向对象编程概述

- 面向对象
 - 将实际问题分解成不同的对象
 - 不同的对象提供不同的服务
 - 对象之间可以传递消息

面向对象编程概述

• 区别

• 一个 C 程序

- 给定一个文件，每行一个记录，包含两列：路名，井盖数
- 定义变量 `int sum = 0;`
- 读入这个文件，每次一行，保存在字符数组 `c` 中
- 在 `c` 中取出第二列，将其转换为整数 `s`
- `sum = sum + s`
- 循环直到文件结束
- 输出 `sum`

• 多辆可识别井盖的车

- 每条马路：是否已经走过？
- 每个井盖：是否已被统计？
- 每辆车：识别了多少井盖？

• 实际问题解决方法 vs 计算机算法

面向对象编程概述

- 面向对象
 - 将实际问题分解成不同的对象
 - 对象向外界提供服务
 - 对象之间可以传递消息
 - 例如
 - 可能对象：马路，井盖，车
 - 服务
 - 马路：通知外界是否访问过
 - 井盖：通知外界是否统计过
 - 车：记录井盖数
 - 消息
 - 井盖 → 车，车 → 井盖，马路 → 车，车 → 马路
- 多辆可识别井盖的车
 - 每条马路：是否已经走过？
 - 每个井盖：是否已被统计？
 - 每辆车：识别了多少井盖？

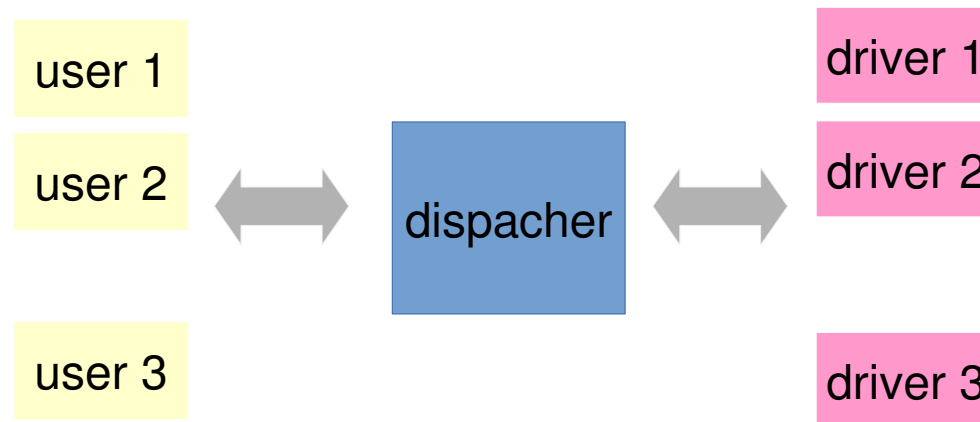
- 例子：打车软件

- 场景：

小李深夜 1 点到达虹桥机场，他打电话给出租车公司，想要订一辆车。公司调度员小马通知了司机小刘，小刘告诉小马自己可以接机，半小时后到。小马通知小李，接机司机正在路上。

- 对象

- 用户，调度者，司机

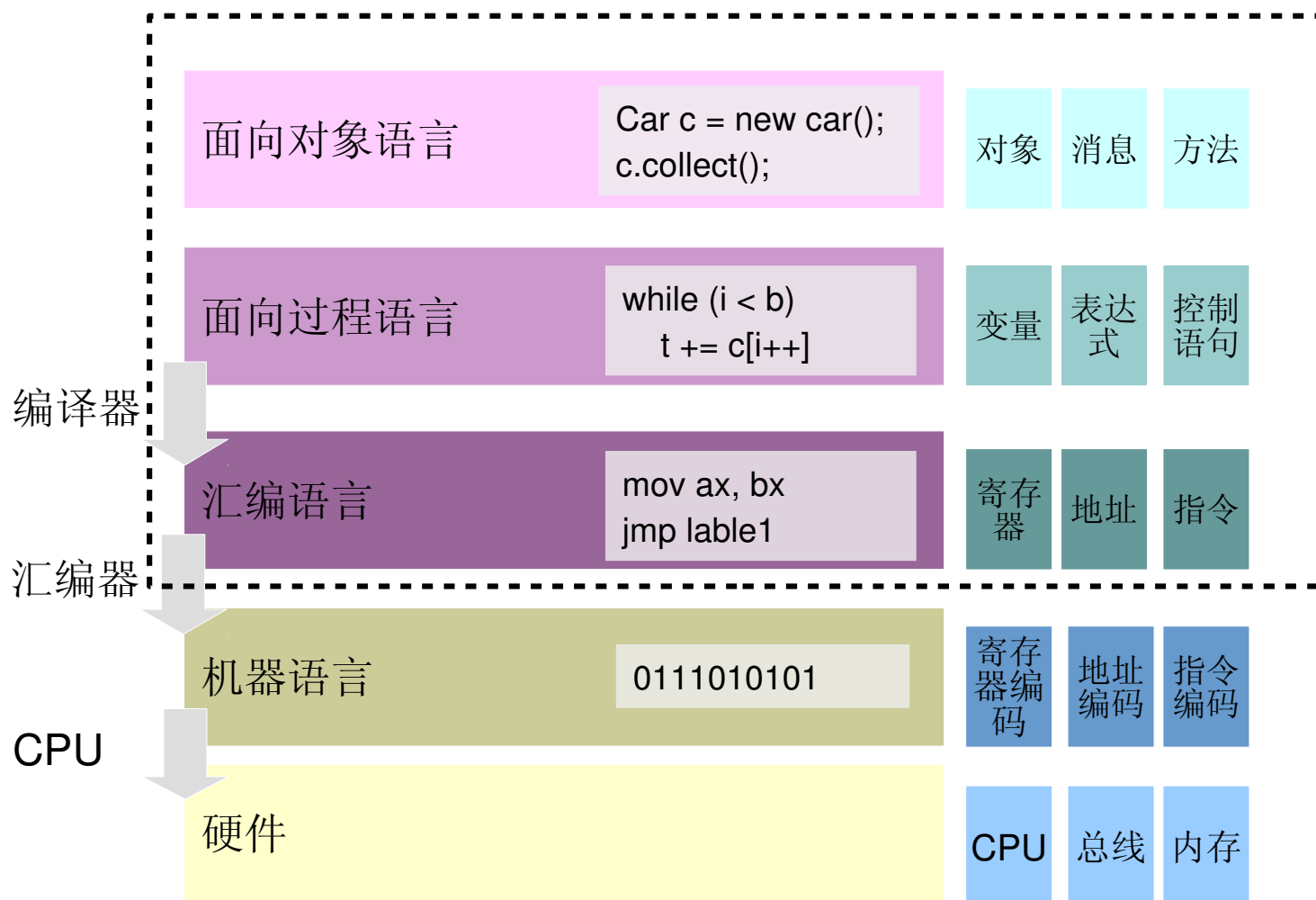


面向对象编程概述

- 面向对象语言
 - 编程语言直接提供了对对象的支持
 - 定义对象，对象提供的服务，消息传递方法
 - 优点
 - 缩短实际问题到计算机算法的距离

面向对象编程概述

- 程序语言的抽象层级



面向对象编程概述

- 面向对象编程要素
 1. 任何事物都是对象
 2. 程序为一些对象间的相互的协作
 3. 一个对象可以包含另一个对象
 4. 每个对象都有类型
 5. 同一类型的对象接收相同类型的消息，提供相同类型的服务

对象的基本要素：状态，行为，类型

面向对象编程概述

- 对象的状态 (state)
 - 每一个对象有自己的状态
 - 井盖对象：是否被访问过？
 - 账户对象：有多少存款？
 - 课程对象：是否正在上课？
 - 程序：一组对象状态的改变
 - 如何改变对象的状态？

面向对象编程概述

- 对象的接口 (Interface)
 - 对象向外界提供的服务
 - 井盖对象 : `visit()` 将其标记为已经访问过
 - 账户对象 : `deposit(int num)` 存入钱款
 - 课程对象 : `start()` 开始课程

面向对象编程概述

- 对象的接口 (Interface)
 - 接口的实现 (Implementation)
 - 隐藏实现细节 (封装, Encapsulation)
 - 井盖对象：如何标记是否被访问？
 - 账户对象：有几位精度？
 - 课程对象：课程名称存储方式？

In any relationship, it's important to have boundaries that are respected by all parties involved

不该看的不看

面向对象编程概述

- 对象的类型 (type, class)
 - 一组行为相同的对象 (仅仅状态不同)
 - 不同的井盖 `class ManholeCover`
 - 不同的账户 `class Account`
 - 不同的课程 `class Lesson`
 - 内置类型：
 - `int, float, double, int[10]`
 - `struct + typedef`
 - 增加新的类型来扩展编程语言

面向对象编程概述

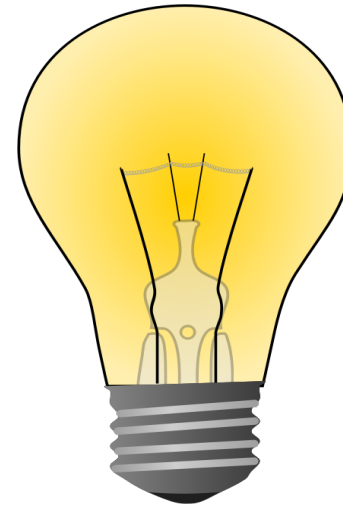
- 对象 vs. 类型

- 类型



Class name

Interface



- 对象

```
Light m = new Light();  
m.on();  
Light n = new Light();  
n.off();
```

面向对象编程概述

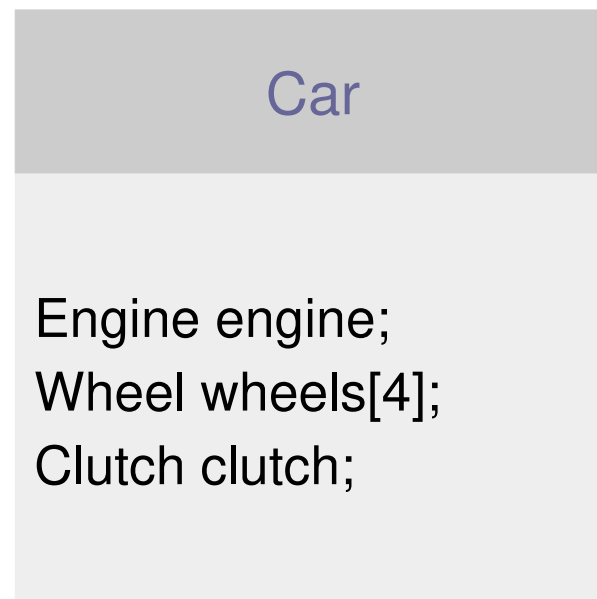
- 利用已有类型如何定义新的类型？
 - 复用 (Reuse classes)

面向对象编程概述

- 利用已有类型如何定义新的类型？

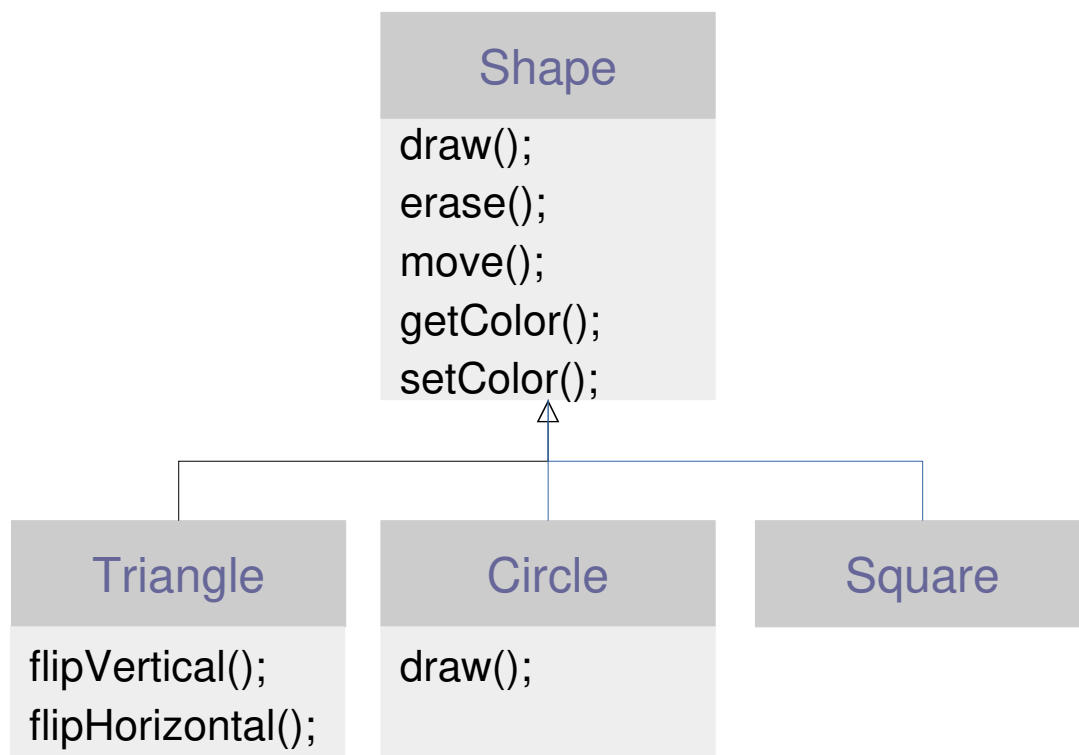
- 组合 (composition) **has-a** 关系

- 引擎类 : `class Engine`
 - 轮胎类 : `class Wheel`
 - 离合器类 : `class Clutch`
 - 汽车类？



面向对象编程概述

- 利用已有类型如何定义新的类型？
 - 继承 (Inheritance) **is-a** 关系

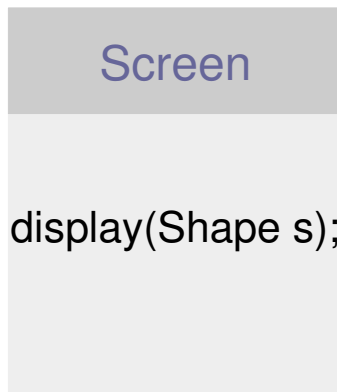


扩展 **Extend**

重写 (**Overriding**)

面向对象编程概述

- 多态 (Polymorphisim)



```
display(Shape s) {  
    s.draw();  
}
```

?

```
Screen s = new Screen();  
Circle c = new Circle();  
s.display(c);
```

面向对象编程概述

- 总结
 - 程序语言的不同抽象
 - 问题空间与计算模型
 - 对象的定义
 - 对象的状态
 - 接口与封装
 - 对象的类型
 - 类的复用：组合，继承，多态

面向对象编程 Java

- Java 简介
- 面向对象编程概述
- 课程介绍

课程介绍

- 课程主页
 - <http://ybwu.org/ecnu-java/16-Spring/index.html>

课程介绍

- 主要内容
 - **Java** 及面向对象编程简介
 - 内置数据与控制结构
 - 类创建与销毁
 - 访问控制
 - 类的复用
 - 多态
 - 接口和内部类
 - 面向对象的 **C** 实现
 - 容器
 - 异常处理
 - 模板
 - **I/O**

课程介绍

- 主要参考书
 - Thinking in Java (4th Edition), Bruce Eckel
 - Java 编程思想 (第四版)
- 阅读
 - Java 文档
 - 其他 Java 教程

课程介绍

- 实验
 - Java 的特点体现在解决实际问题上
 - 实验会超前（非线性）

课程介绍

- 实验原则
 - 不接收迟交作业
 - 请尽早动手，请尽早动手，请尽早动手
 - 不允许抄袭
 - 一旦发现，无论抄袭或被抄袭都判为 0 分
 - 鼓励小组讨论

Start Early!
No Cheating!

课程介绍

- 作业提交规范
 - 作业文件命名方式：
 - 学号_姓名.zip
 - 对实验中的每个子问题，建立一个文件夹，包含相应源代码以及使用说明 (**README**).
 - 源代码为你的 **Java** 程序 (仅包含 **.java** 文件，不包含 **.class** 文件)
 - **README** 为一个文本文件，说明如何使用你的程序 (例如：参数如何给出，每个参数含义)

课程介绍

- 联系方式
 - 教师
 - 吴苑斌, ybwu@cs.ecnu.edu.cn, 闵行信息楼 610
 - 助教
 - 李晨瑞, lcrr2011@163.com
- 答疑时间
 - 周二, 1:00am – 3:00am, 理科楼 B803

课程简介

- 实验一
 - 下周一晚 9 点前发送邮件到助教邮箱