

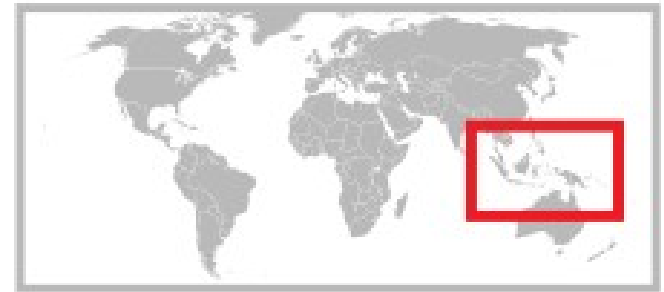
# 面向对象编程 Java

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 17





译：Equim

来介绍一下

这是从隔壁东京Node学园转来代码学园的JavaScript同学。

坐在之前  
ActionScript同学的位置就好了

初、  
初次见面...

我叫JavaScript。  
叫我JS就好。  
请多多关照...

"Java"Script ...?

※译注：从外形上看，老师很可能是  
クラウドピア窓辺(克劳迪亚·窗边)，即Azure娘。

※译注：东京Node学园在现实中是由「Node.js日本User Group」主催的Node.js学习会，官网：<https://nodejs.connpass.com/>

※译注：东京Node学园在现实中是由「Node.js日本User Group」主催的Node.js学习会，官网：<https://nodejs.connpass.com/>

你也叫Java呢。

我是『Java』。  
担任这座学园的学生会长。

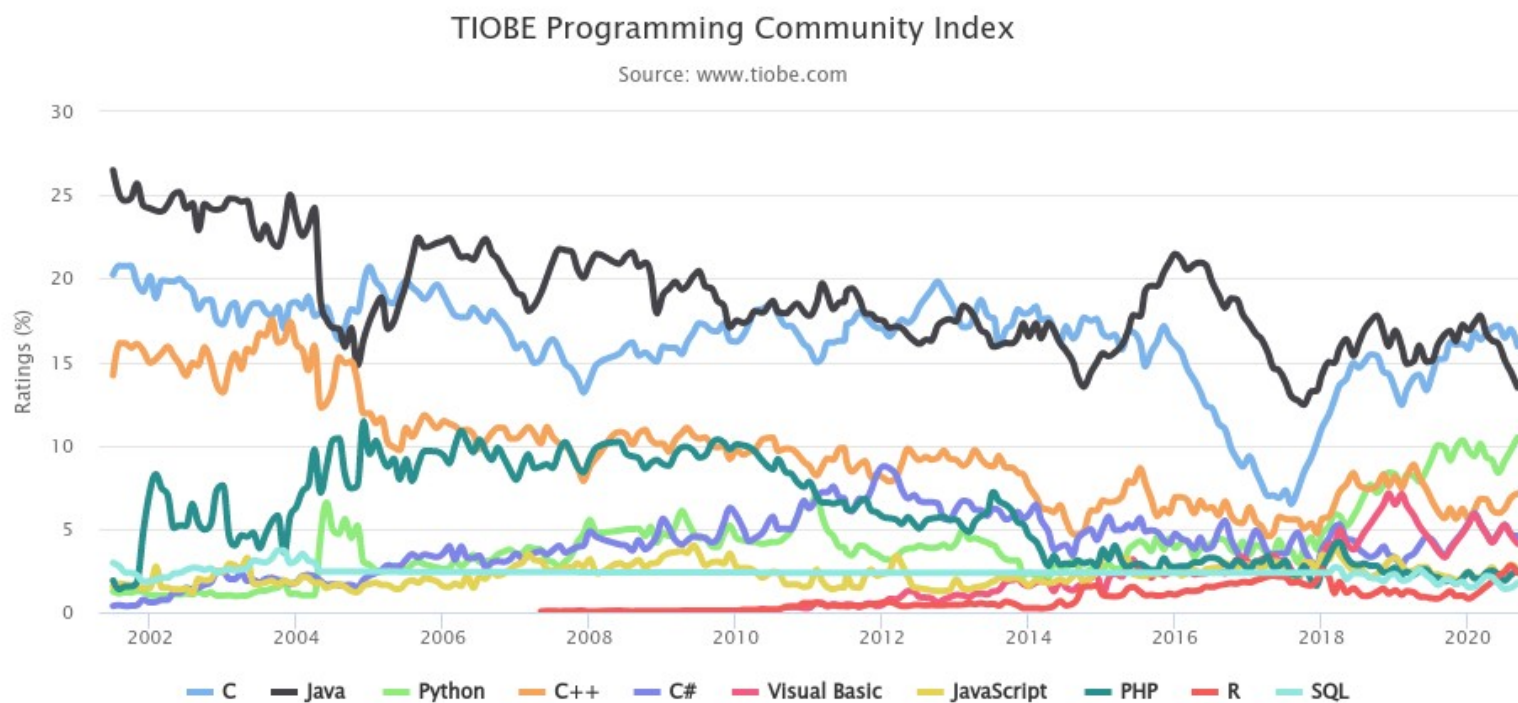
※译注：Java酱的臂章「型」应该是在暗指Java的基本类型和引用类型

# 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！
- C 语言是否可以？
- Java 如何实现 WORA？
  - Java 虚拟机 (Java Virtual Machine, JVM)
  - Java bytecode

# 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





※译注：关于C语言酱的设定，天然：更接近汇编语言；不可思议：能通过指针完成很多复杂的操作；Destroyer：对内存的操作稍有不慎会有隐患

# 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）

# 第一个 Java 程序



Hello



# Java 简介

- 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

- Python 语言实现

- 主函数：main ?
- 标准输出函数：print
- 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

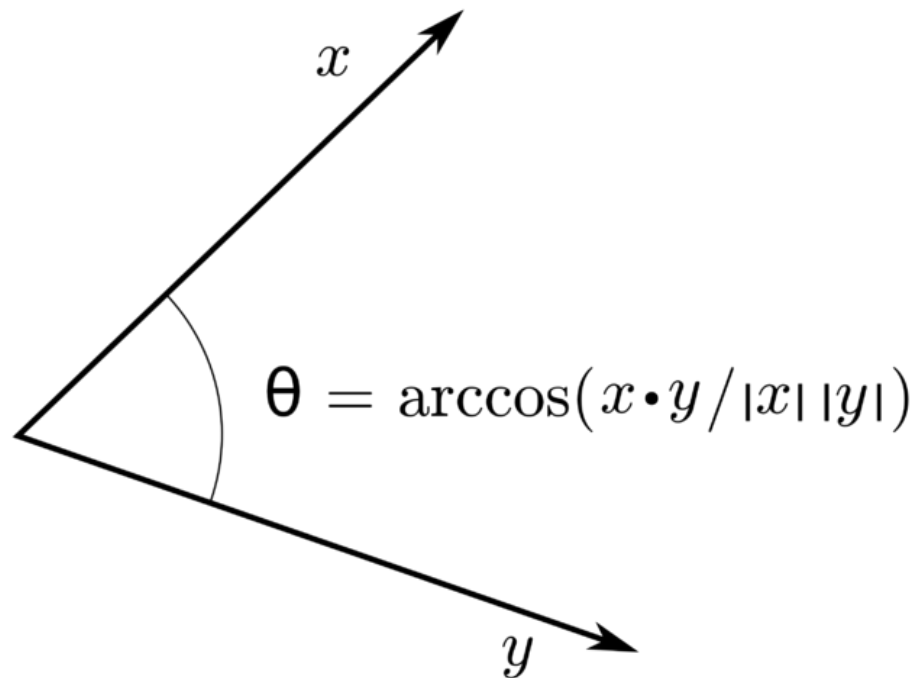
- Java 简介
- 面向对象编程概述
- 课程介绍

# 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：计算两个向量的内积



# 面向对象编程概述

- 问题：计算两个向量的内积
  - 一个 C 程序
    - 定义函数
    - `double inner_prod(double x[], int x_len, double y[], int y_len);`
    - 实现函数
      - 循环，累加
  - 观察：
    - 向量：使用数组实现
    - 参数：每个数组都需要跟随一个长度



# 面向对象编程概述

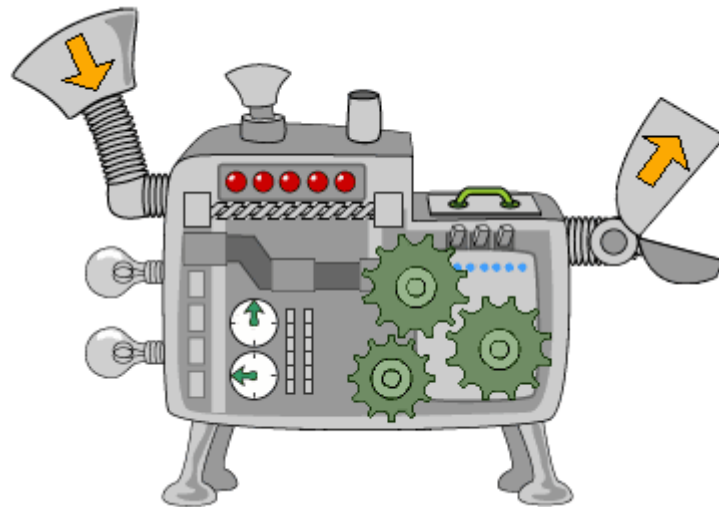
- 问题：计算两个向量的内积
  - 假设可以为**向量**增加一些功能：
    - 功能 1: 告诉外界自己的长度
    - 功能 2: 当用户输入  $i$  时，返回它的第  $i$  维元素
    - 功能 3: 当用户输入另一个向量时，返回它们的内积

# 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

- 例如：向量内积问题
  - 可能对象：向量
  - 服务
    - 告诉外界自己的长度
    - 当用户输入  $i$  时，返回它的第  $i$  维元素
    - 当用户输入 另外一个向量时 返回它们的内积



# 面向对象编程概述

- 对比 C 语言的实现
  - 两个 **double 类型数组**间的关系
  - 两个**名为向量的对象**间的关系

- 另一个例子：打车软件

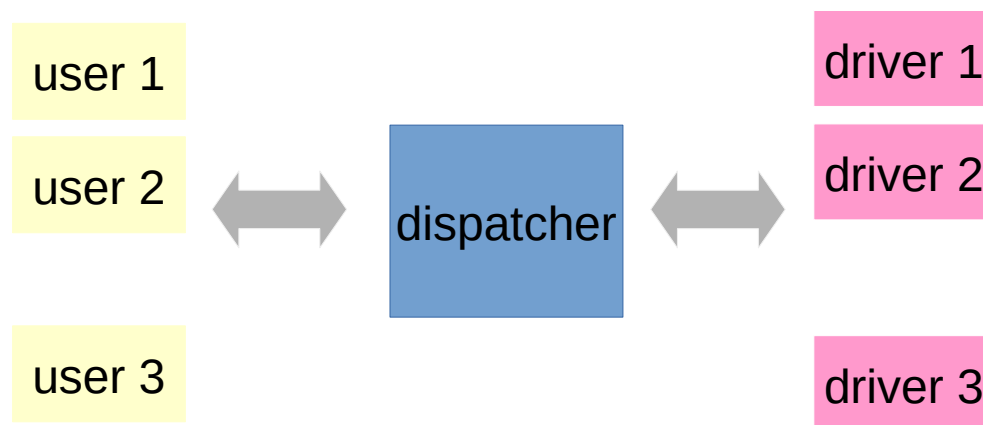
- 场景：

小李深夜到达虹桥站，他打电话给出租车公司订一辆车。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。小马通知小李，接机司机正在路上。

- 对象

- 消息传递

- 服务



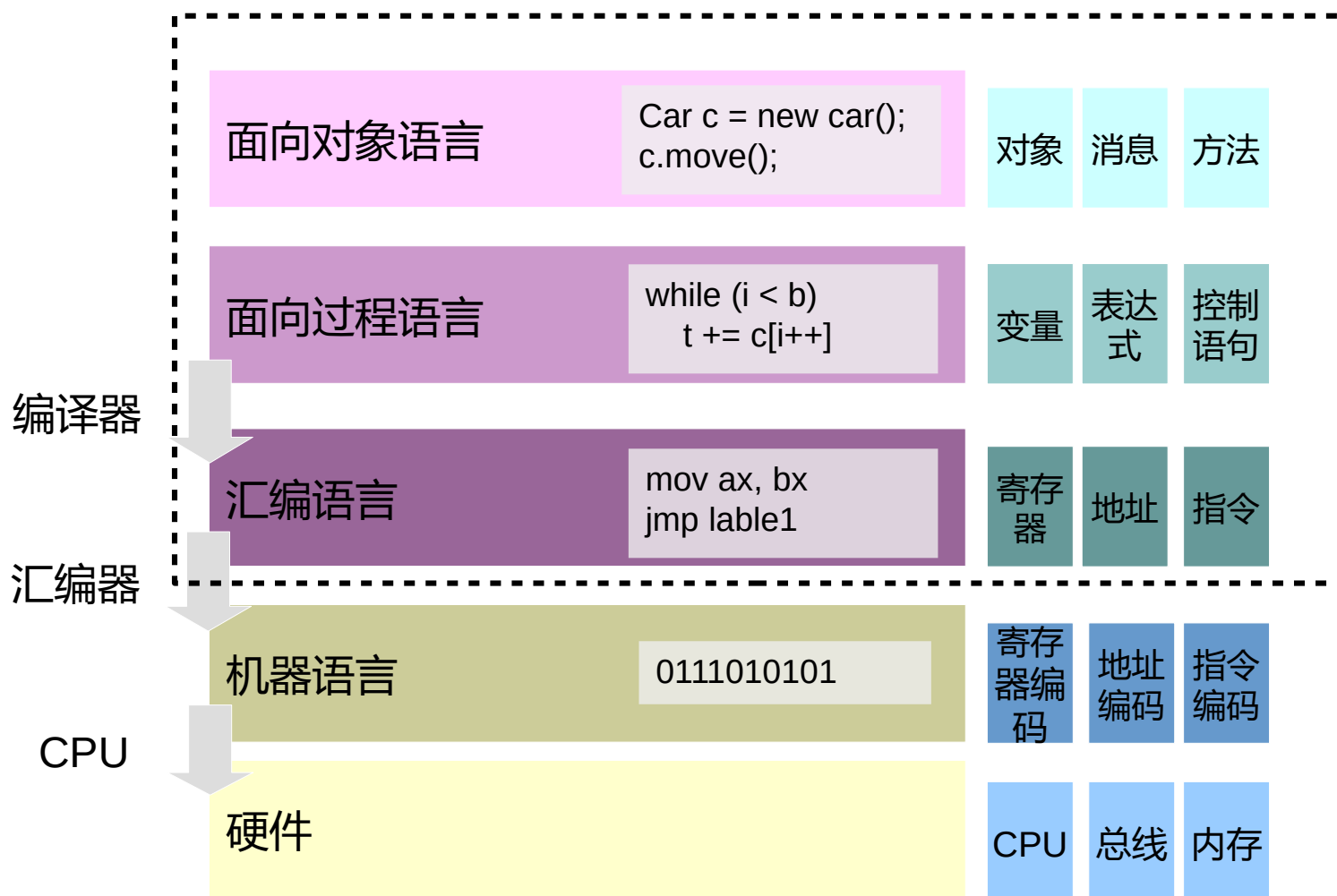
# 面向对象编程概述

- 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”是一些对象间的相互协作
- 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

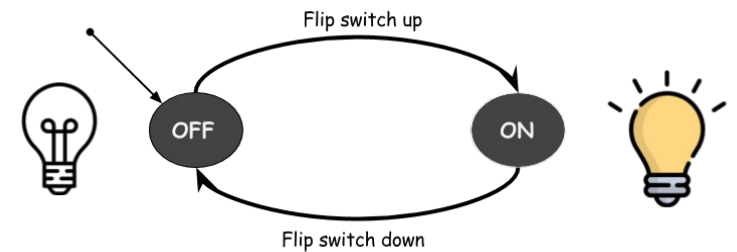


对象的基本要素：状态，行为，类型

# 对象的基本要素：状态，行为，类型

- 对象的**状态** (state)

- 每一个对象有自己的状态
  - 向量对象：当前的第  $i$  维是什么？
  - 账户对象：有多少存款？
  - 灯泡对象：开 or 关？
- 程序：一组对象状态的改变
- 如何改变对象的状态？



# 对象的基本要素：状态，**行为**，类型

- 对象的**行为**

- 对象向外界提供的服务，
  - 向量对象：set(int i, double v) 设置第 i 维元素为 v
  - 账户对象：deposit(int num) 存入钱款 num
  - 灯泡对象：turn\_on() turn\_off(), 开灯，关灯
- 称为“接口” (interface)

# 对象的基本要素：状态，**行为**，类型

- 接口

- 接口的实现 (Implementation)
- 隐藏实现细节 (封装, Encapsulation)
  - 向量对象：具体实现方式 (数组 or 链表)
  - 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**

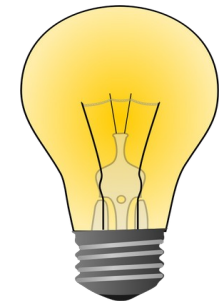
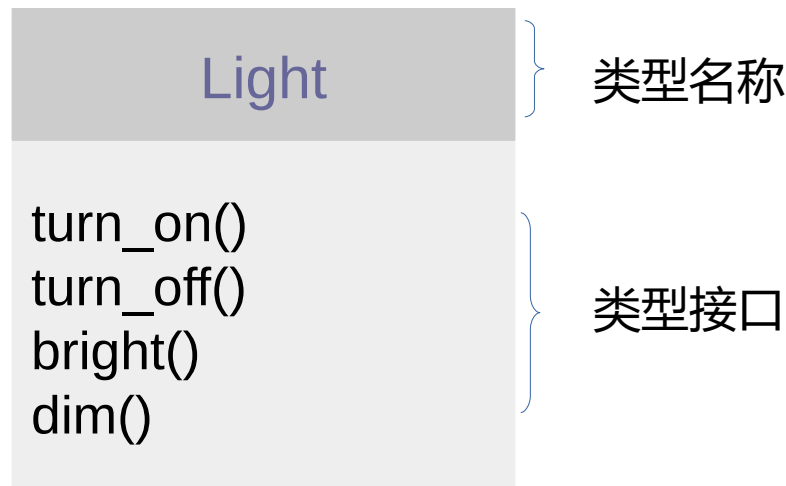
# 对象的基本要素：状态，行为，类型

- 对象的**类型** (type, class)
  - 一组行为相同的对象（仅状态不同）
  - 例如：
    - 向量对象  $x$ , 表示向量  $[1, -1]$
    - 向量对象  $y$ , 表示向量  $[0, 4, 5]$
    - 对象  $x, y$  是不同的. 但它们的行为相同（提供相同的服务，接收相同的消息）
    - 属于有同样的类型 **class Vector**
  - 类似
    - 不同的账户对象属于类型 **class Account**.
    - 不同的课程对象属于类型 **class Lesson**.
  - 类比于 C 语言的变量类型
    - `int, float, double, int[10]`
    - `struct + typedef`: 增加新的类型来扩展编程语言

# 对象的基本要素：状态，行为，类型

- 对象 vs. 类型

- 类型



- 对象

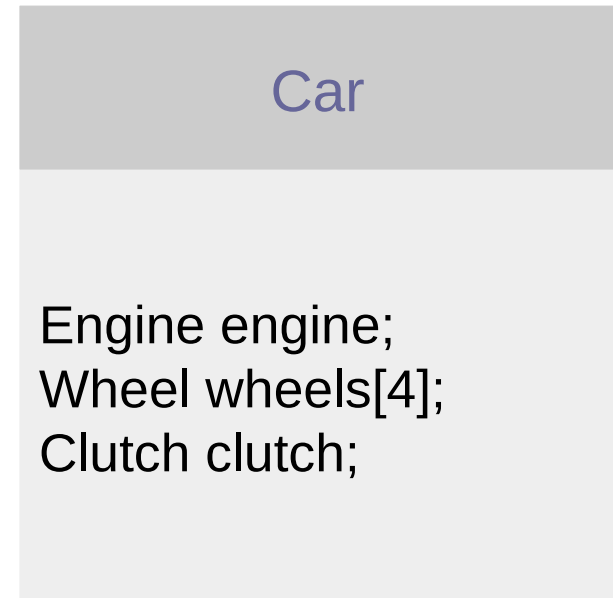
```
Light m = new Light();  
m.turn_on();  
Light n = new Light();  
n.turn_off();
```

# 对象的基本要素：状态，行为，**类型**

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

# 对象的基本要素：状态，行为，**类型**

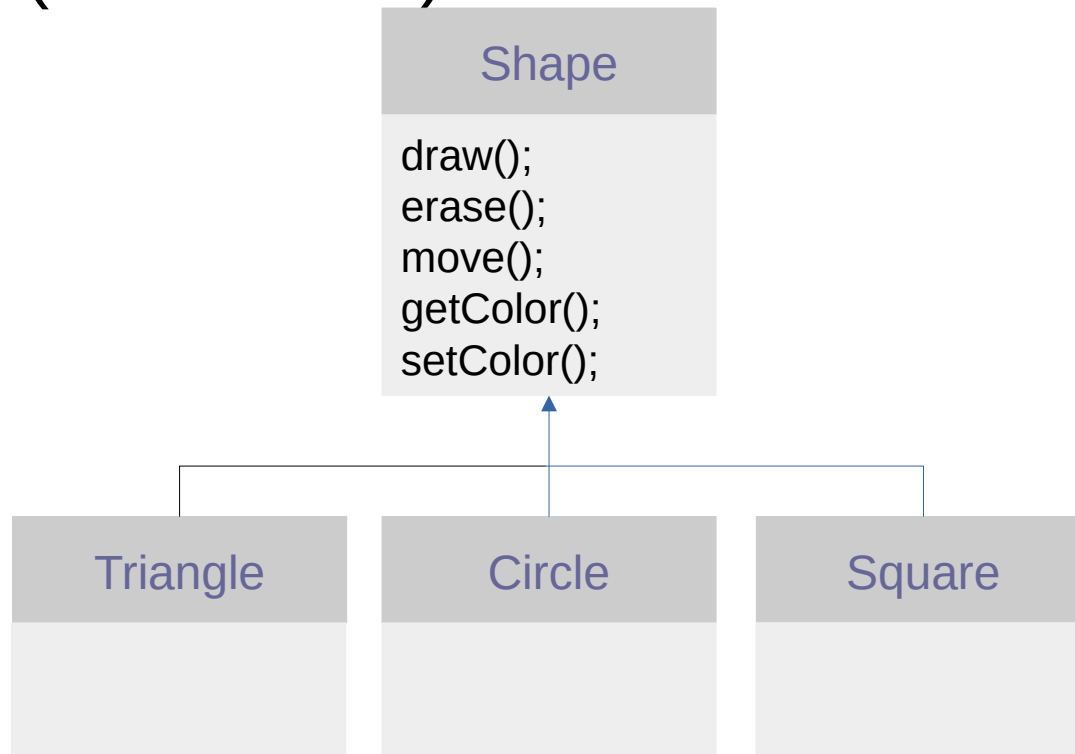
- 利用已有类型如何定义新的类型？
  - 组合 (composition) **has-a 关系**
    - 引擎类：class Engine
    - 轮胎类：class Wheel
    - 离合器类：class Clutch
    - 汽车类？





# 对象的基本要素：状态，行为，**类型**

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



# 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

# 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
  - <https://github.com/ecnu-oop-java/ecnu-oop-java-up/23-Spring/README.md>

# 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
  - 容器
  - 异常处理
  - 模板
  - I/O

# 课程介绍

- 主要参考书
  - Thinking in Java (4<sup>th</sup> 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

- 评分
  - 期末考试（闭卷）
  - 课堂小测（随堂）

# 课程介绍

- 实验
  - 规定内时间完成
  - **实验会超前**



# 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
  - 作业提交邮箱：
    - ecnu\_java\_2023@163.com
  - 作业文件命名方式：
    - 实验编号\_学号\_姓名.zip
    - 例如：Lab1\_10205102000\_陈东.zip
  - 提交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, 理科楼 B911

- 助教

- 杨润东, 10205102454@stu.ecnu.edu.cn
    - 刘诺维, 10191900446@stu.ecnu.edu.cn

- 答疑时间

- 周三 13:00 – 15:00, 理科楼 B911

# 课程简介

- 实验一

- 下周三 (3 月 8 日) 晚 9 点前发送邮件到  
ecnu\_java\_2023@163.com