

# 面向对象编程基础

宿宝臣 <subaochen@126.com>

山东理工大学

October 27, 2020

# 概述

- 1 “类”的由来和表达
- 2 类和对象
- 3 类的组织
- 4 类的层次
- 5 类的保护

## 两种分析思路

以ATM机存款为例：

- 竖切，时间线是关键，表达方式：流程图(flowchart)
- 横切，相互关系是关键，表达方式：用例图(use case)

# struct是对C语言数据类型的扩展

- 什么是数据类型？
- 简单数据类型？
- C语言的复杂数据类型：struct
- struct point

# class是对Java数据类型的扩展

- Java的简单数据类型
- String是简单数据类型吗？
- struct的进一步扩展：class
- class point

# Java如何定义类:class

- 关键字：class
- 语法规范
- 属性(property)=field/variable
- 方法(method)=function

# 对象=实例

- 类：对象=1:n
- object=instance

# 对象的创建

- new
- instanceof
- 构造方法的基本概念
- 构造方法的几种形式和演示



# 属性和方法的引用

- 运算符.

# 类属性和类方法

- 为什么要引入类属性和类方法？
- 类属性和类方法的表达：static
- 类属性和类方法的引用
- 类属性和类方法的使用限制：实例属性和方法的引用冲突
- 演示：

# 实战

- Point/Rectangle/Circle/Draw
- - .运算符可以用来访问属性和执行方法，就像struct中访问成员分量
- 不断提高自己的抽象思维能力

# 类的分类存储：包

- 包(package)=目录

# 包的命名原则

- 保证项目内的唯一性
- 常见：逆序域名，保证了全球唯一性

## 包的导入和声明

- import(类比C的include编译预处理命令)
- 导入单个类
- 导入多个类
- 导入类属性

# 继承的引入

- Shape类的引入
- 继承：更高的抽象，找出现有类的公共特征（属性、方法）
- super关键字

## 继承的优点和缺点

- 优点：更高的抽象，精简的代码，构造树状类层次结构
- 缺点：高度依赖



## 方法覆盖（override）和重载(overload)

- override/@override:父子之间
- overload:类内部

# 为什么要保护属性和方法？

- 保护：有限制的访问
- 属性和方法的作用范围的进一步延伸

# 如何保护？

- 语法上的限制

修饰词	Class	Package	Subclass	World
public	√	√	√	√
protected	√	√	√	X
( default )	√	√	X	X
private	√	X	X	X

# 保护详解

- 保护种族（包）：(default)，即无须声明任何访问限制

# 保护详解

- 保护种族（包）：(default)，即无须声明任何访问限制
- 保护家族（父类）：protected

# 保护详解

- 保护种族（包）：(default)，即无须声明任何访问限制
- 保护家族（父类）：protected
- 保护自己：private

## 保护详解

- 保护种族（包）：(default)，即无须声明任何访问限制
- 保护家族（父类）：protected
- 保护自己：private
- 最佳实践：private->(default)->protected->public，让代码更安全

# 保护实战

- 实验包范围的保护
- 实验父类属性和方法的保护



## 专题：构造方法的保护

- 了解不同保护范围的构造方法的意义