

《Java语言程序设计》

-美 梁勇 摘录

基础：

源文件名必须和公共类名完全相同

每个java程序都是以类的声明开始的，每个文件必须要有一个类

一个例子：

eclipse里面：建立一个project 名字为demo1, 建立一个类Welcome

目录结构自动变成：demo1/ src /demo1 / Welcome.java

编译运行：在demo1/ src /demo1下 `javac Welcome.java`

然后： `cd ..`

`java Welcome`

在终端输入jshell, 可以快速验证计算结果

`/vars` 打印所有已定义的变量

`int a = 3`

`int b = 4`

`Math.pow(a, b)`

两个整数相除的结果还是整数： $5/9 = 0$

类型转换：显式和隐式 `int a = 126; byte b = (byte) a`

C的char占1个字节，java占2个字节

`Character.isDigit()` 只对0-9返回true

方法：驼峰命名法； 类：首字母大写

方法重载：

可以同时定义名字相同的函数，但是参数或者参数精度必须不同。编译器会自动匹配精度，选择最合适的那个

数组复制：

`list1 = list2;` 之后：list1 指向和list2 一样的数组，list1 原先指向的数组被垃圾回收

与C不同的是：不能在main函数里面定义和创建函数

pass-by-value: 按值传递, 例: 一般数据类型

pass-by-sharing: 引用传递, 传递的是同一个对象; 例: 数组, 一改全改

java将对象存储在堆当中

可变长参数列表:

将可变数量的参数传递给方法, 并将其视为数组, 但一个方法里面最多一个, 并且为参数列表里面的最后一个

例如: `public static void printM(int len, double... numbers)`

二维数组传参, 不需要指明第一维的长度

对象和类

在一个类中简单地加入main方法来测试该类

构造方法与类同名, 完成的是初始化工作, 可重载。有无参构造方法, 默认构造方法

引用变量:

`Circle myCircle = new Circle()`其中myCircle存放的是对Circle对象的引用

对象成员, 数据域, 方法, 实例方法是非静态定义的, 不能通过类来调用。对引用变量来说, 数据域有默认初始化, 函数内部没有默认初始化, 打印会报错

对象类型赋值同引用类型赋值本质一样

类变量-静态变量- `static int var`

静态方法 `static int getvar(){}`

静态: 可直接被类调用

Math中的所有方法都是静态的

静态变量和静态方法都用classname来调用, 这样易读

静态方法不能调用实例方法和访问实例数据域

如何确定定义? 如果不依赖具体的实例, 则定义为静态方法

类中常量为所有实例共享: `final static double PI = 3.14`

可见性修饰符

public 能被任何其他的类访问

默认可被同一个包中任何一个类访问

private 私有, 只能类内访问, 只能用在类内的成员上

包用以组织类

`package packageName` 定义类所在的包，如没有，使用的是默认包

区分局部变量和类的成员

`private Math(){}` 可以防止用户创建Math的实例

UML

类的UML符号：

下划线：静态

-: private 修饰符

+: public 修饰符

数据封装：设置访问器和修改器，防止数据篡改

对象

对象作参数，是按值传递，本质上是引用传递

存在对象数组

不可变对象和类

要使类不可变：

1. 所有数据私有
2. 没有修改器方法
3. 没有返回一个指向可变数据域的引用的访问器方法

作用域

无论在何处声明，类变量的作用域都是整个类。类变量包括实例变量和静态变量

this引用

this即本对象

`this.radius = radius` (函数参数与静态数据域重名) 或者 `this(1.0)` 使用另一个重载函数