# static 修饰变量

重点

可以用来修饰变量、类、修饰方法

static class 类名 {

}

static void 方法名() {

}

static int 变量;

限制：static 不能是方法内的局部变量

注意：

1） 如果static变量没有初始化，那么会有一个默认值

byte short int long 0

char '\u0000'

boolean false

float double 0.0

其他类型 null (没有地址，不指向任意一块内存空间) 空

相对来讲，局部变量没有默认值，如果不赋值，不能使用

1. 全局变量和局部变量可以命名重复，当命名有冲突时，采用就近原则（局部变量优先）
2. 命名有冲突时，仍要使用全局变量，使用： 类名.全局变量名

# java的面向对象编程

Object Oriented Programming (OOP)

使用java语言对 现实世界中的事物进行抽象, 特征：封装、继承、多态

String name1 = "张三";

int age1 = 18;

int score1 = 90;

String name2 = "李四";

int age2 = 20;

int score2 = 70;

事物 =》 学生 =》 对学生进行抽象 =》 进行归类 =》 变为java的class

class Student {

String name; // 从属于学生的特性 (姓名)

int age; // 从属于学生的特性 (年龄)

int score; // 从属于学生的特性 (成绩)

}

这段代码，只是定义了学生，不是具体的某一个学生。

要体现具体学生，需要下面实例化的过程

// 新建

instance object

Student stu1 = new Student(); // 结果称之为“实例” 或 “对象”

stu1.name = "张三";

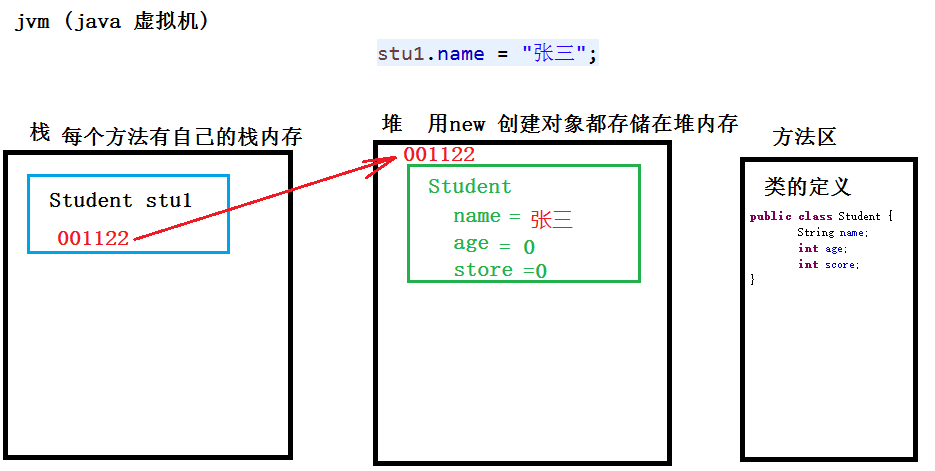
stu1.age = 18;

stu1.score = 90;

没用面向对象之前，变量之间没有联系； 使用了面向对象以后，这些变量通过“对象”联系在了一起

## 类与对象的内存结构

重点



## NULL 值

非基本类型的变量，都可以赋值为null

非基本类型的变量，内部存储的都是地址(引用), null 表示没有地址

Student stu1 = null; // stu1 没有指向任何一个地址

要注意空指针异常：java.lang.NullPointerException 空指针异常

当变量取值为null时，调用它的成员变量、成员方法时就会报这个错误

## 构造方法

重点

作用： 给类中的成员变量执行初始化操作

定义：方法名与类名一致，不需要定义返回值，不能是static

## 默认构造方法

重点

如果没有提供自定义的构造方法，每个class 也会有一个默认构造方法(无参构造)

public 类名(){

}

一旦提供了一个任意的自定义构造，默认构造方法就会消失

构造方法可以重载

## 成员变量 属于对象

## 成员方法 属于对象

格式：　不能有static

public 返回值 方法名(参数列表) {

// 方法内存

}

调用格式：

对象.方法名(实参)

对象引用.方法名(实参)

成员方法执行，可能会受到成员变量的影响

如果成员变量，与成员方法内的局部变量有命名冲突，仍然采用“就近原则”

如果有了冲突，还想使用成员变量 this.成员变量名

## This

重点

只能用在构造方法，或是成员方法当中

this代表当前对象自身

用法1：

this.成员变量

this.成员方法()

当没有命名冲突的情况下，可以省略this

用法2：this用来构造方法

this(参数)

必须在构造之间进行调用, 其他成员方法不能调用this(参数)构造方法

如果有多行语句，this(参数)必须位于第一行

构造方法里可以调用其他成员方法

## 封装

1. 把复杂的逻辑隐藏起来，暴露简单的调用方法, 或是保护重要的成员变量
2. 类与类之间的调用可以通过访问修饰符加以控制

重点

public 公共的 修饰的变量和方法，在本包和其他包都能被访问

protected 受保护的 被修饰的变量和方法只能被本包和子类访问

不加任何修饰符 (default) 默认的 被修饰的变量和方法只能在本包内被访问

private 私有的 只能是本类进行访问

从宽到窄

public > protected > default > private

修饰类： 只能是public 或是 default

default 修饰类，在本包内可以用，出了包，就被隐藏

public 修改类，所有包都可以用（一个文件中只能有一个public类，public类名必须和文件名一样）

作业：

1. 有以下代码

class MyClass{

int value;

}

public class TestRef{

public static void main(String args[]){

int a = 10;

int b = a;

b ++ ;

System.out.println(a);

MyClass mc1 = new MyClass();

mc1.value = 10;

MyClass mc2 = mc1;

mc2.value ++;

System.out.println(mc1.value);

}

}

请写出编译运行后的结果:10,11.

1. 有以下代码

class ClassA{

int value = 10;

}

public class TestReturnRef{

public static void main(String args[]){

ClassA ca = new ClassA();

ca = getObject();

ca = getObject();

ca = getObject();

System.out.println(ca.value);

}

public static ClassA getObject(){

ClassA newObject = new ClassA();

newObject.value += 10;

return newObject;

}

}

编译运行TestReturnRef 程序，结果为? 20

3. 有以下代码,请选择正确答案：

class ClassA{ public ClassA(){ System.out.println("ClassA()"); } }

class ClassB{ public ClassB(){ System.out.println("ClassB()"); } }

class ClassC{

ClassA a = new ClassA();

ClassB b;

public ClassC(){

System.out.println("ClassC()");

b = new ClassB();

}

}

public class TestConstructor{

public static void main(String args[]){

ClassC cc = new ClassC();

}

}

1. 编译不通过
2. 输出ClassA() ClassB() ClassC()
3. 输出 ClassA() ClassC() ClassB()
4. 输出 ClassC() ClassB() ClassA()

答案：c

4. 请选择正确答案：

class ClassA {

int value;

}

public class Test4 {

public static void main(String args[]) {

int value = 10;

changeInt(value);

System.out.println(value);

ClassA ca = new ClassA();

ca.value = 10;

changeObject(ca);

System.out.println(ca.value);

}

public static void changeInt(int value) {

value++;

}

public static void changeObject(ClassA ca) {

ca.value++;

}

}

编译运行Test4时，结果是

1. 编译出错

B. 输出 10 11

C. 输出 10 10

D. 输出 11 11

B

5.请选择正确答案：

class ClassA {

int value;

}

public class Test4 {

public static void main(String args[]) {

ClassA ca = new ClassA();

changeValue(ca);

System.out.println(ca.value);

changeRef(ca);

System.out.println(ca.value);

}

public static void changeValue(ClassA ca) {

ca.value = 100;

}

public static void changeRef(ClassA ca) {

ca = new ClassA();

ca.value = 200;

}

}

编译运行Test4，结果为

1. 编译不通过
2. 输出100 200
3. 输出100 100
4. 输出0 200

答案：C

第二次输出的依然是改动前的值，改动后的值维为200，但是未输出。