周记

1. 数组
2. 数组的概念

**数组**（Array）是有序的元素序列。   若将有限个类型相同的变量的[集合](https://baike.baidu.com/item/%E9%9B%86%E5%90%88/2908117" \t "https://baike.baidu.com/item/_blank)命名，那么这个名称为数组名。组成数组的各个变量称为数组的分量，也称为数组的元素，有时也称为[下标变量](https://baike.baidu.com/item/%E4%B8%8B%E6%A0%87%E5%8F%98%E9%87%8F/12713827" \t "https://baike.baidu.com/item/_blank)。用于区分数组的各个元素的数字编号称为下标。数组是在[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952" \t "https://baike.baidu.com/item/_blank)中，为了处理方便， 把具有相同类型的若干元素按有序的形式组织起来的一种形式。 这些有序排列的同类数据元素的集合称为数组。

数组是用于储存多个相同类型数据的集合。

1.1语法结构

dataType[] arrayRefVar; // 首选的方法

dataType arrayRefVar[]; // 效果相同，但不是首选方法

1.2示例

int[] array={12,23,34,45} // 这是int类型的一维数组

double[][] array1=new double[23][34]//这是一个double类型的二维数组

1.2.1输出数组里面的数值

//一般会用到for循环遍历的方法

public class TestArray {

public static void main(String[] args) {

double[] myList = {1.9, 2.9, 3.4, 3.5};

// 打印所有数组元素

for (int i = 0; i < myList.length; i++) {

System.out.println(myList[i] + " ");

}

｝

｝

1.2.2数组的扩容

初始化数组要求指定数组容量，但在许多情况下需要扩充数组容量。

有两种方法可以实现数组容量扩容：

public class Demo1 {

static int []array=new int [20];

public static void main(String[] args){

array=addLengthArray(array);

for(int i = 0;i<array.length;i++){

array[i]= i;

System.out.println(array[i]);

}

}

public static int[] addLengthArray(int[] array){

int[ ] newArray = new int [array.length\*2];

//将array数组从0位置至array.length位置，复制到newArray数组0位置到array.length位置。

System.arraycopy(array,0,newArray,0,array.length);

return newArray;

}

}

1.2.3数组的合并

数组合并也是经常用到的一种方式

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class Main {

public static void main(String args[]) {

String a[] = { "A", "E", "I" };

String b[] = { "O", "U" };

List list = new ArrayList(Arrays.asList(a));

list.addAll(Arrays.asList(b));

Object[] c = list.toArray();

System.out.println(Arrays.toString(c));

}

}

1. 面向对象
2. 概念

面向[对象](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1" \t "https://baike.baidu.com/item/_blank)(Object Oriented)是[软件开发方法](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E6%96%B9%E6%B3%95/971447" \t "https://baike.baidu.com/item/_blank)。面向[对象](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1" \t "https://baike.baidu.com/item/_blank)的概念和应用已超越了[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952" \t "https://baike.baidu.com/item/_blank)和软件开发，扩展到如[数据库系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%B3%BB%E7%BB%9F/215176" \t "https://baike.baidu.com/item/_blank)、交互式界面、应用结构、应用平台、[分布式系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F/4905336" \t "https://baike.baidu.com/item/_blank)、[网络管理](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%AE%A1%E7%90%86/5903609" \t "https://baike.baidu.com/item/_blank)结构、CAD技术、[人工智能](https://baike.baidu.com/item/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD/9180" \t "https://baike.baidu.com/item/_blank)等领域。面向对象是一种对现实世界理解和抽象的方法，是计算机编程技术发展到一定阶段后的产物。

常常有一种说法叫做万物皆对象

1. 特性

面向对象包含三大特征，分别是

**封装**：隐藏对象的属性和实现细节，仅对外提供公共访问方式，将变化隔离，便于使用，提高复用性和安全性。  
 **继承**：提高代码复用性；继承是多态的前提。  
 **多态**：父类或接口定义的引用变量可以指向子类或具体实现类的实例对象。提高了程序的拓展性。

2.1面向对象的组成与结构

主要由基本类（主类）和对象类组成

对象由各个方法（行为）与类的属性组成，对象类调用主类的属性以及方法，

2.1.1一个类可以包含以下类型变量：

****局部变量****：在方法、构造方法或者语句块中定义的变量被称为局部变量。变量声明和初始化都是在方法中，方法结束后，变量就会自动销毁。

**成员变量**：成员变量是定义在类中，方法体之外的变量。这种变量在创建对象的时候实例化。成员变量可以被类中方法、构造方法和特定类的语句块访问。

**类变量**：类变量也声明在类中，方法体之外，但必须声明为 static 类型。

## 2.1.2构造方法

每个类都有构造方法。如果没有显式地为类定义构造方法，Java 编译器将会为该类提供一个默认构造方法。

在创建一个对象的时候，至少要调用一个构造方法。构造方法的名称必须与类同名，一个类可以有多个构造方法。

下面是一个构造方法示例:

public class Dog{

public Dog(){

}

public Dog(String name){

// 这个构造器仅有一个参数：name

}

}

2.2.3创建对象

对象是根据类创建的。在Java中，使用关键字 new 来创建一个新的对象。创建对象需要以下三步：

**声明**：声明一个对象，包括对象名称和对象类型。

**实例化**：使用关键字 new 来创建一个对象。

**初始化**：使用 new 创建对象时，会调用构造方法初始化对象

下面是一个创建对象的例子

public class Puppy{

public Puppy(String name){

//这个构造器仅有一个参数名字

System.out.println("小狗的名字是 : " + name );

}

public static void main(String[] args){

// 下面的语句将创建一个Puppy对象

Puppy myPuppy = new Puppy( "大黄" );

}

}

2.2 this关键字

this关键字除了可以引用变量或者构造方法之外，还有一个重大的作用就是返回类的引用。如在代码中，可以使用return this，来返回某个类的引用。此时这个this关键字就代表类的名称。

#### 2.2.1 引用成员变量

Public Class Student {

String name; //定义一个成员变量name

private void SetName(String name) { (局部变量)name

this.name=name; //将局部变量的值传递给成员变量

}

}

#### 2.2.2调用类的构造方法

public class Student { //定义一个类，类的名字为student。

public Student() { //定义一个方法，名字与类相同故为构造方法

this(“Hello!”);

}

public Student(String name) { //定义一个带形式参数的构造方法

}

}