加餐一讲用一篇文章带你了解专栏中用到的所有Java语法

尽管说设计模式跟编程语言没有直接关系，但是，我们也无法完全脱离代码来讲设计模式。我本人熟悉的是Java语言，所以专栏中的代码示例我都是用Java语言来写的。考虑到有些同学并不熟悉Java语言，我今天用一篇文章介绍一下专栏中用到的Java语法。

如果你有一定的编程基础，熟悉一门编程语言，结合我今天讲的Java语法知识，那看懂专栏中的代码基本不成问题。

如果你熟悉的是C/C++、C#、PHP，那几乎不用费多大力气，就能看懂Java代码。我当时从C++转到Java，也只看了一天的书，基本语法就全部掌握了。

如果你熟悉的是Python、Go、Ruby、JavaScript，这些语言的语法可能跟Java的区别稍微有些大，但是，通过这篇文章，做到能看懂也不是难事儿。

好了，现在，就让我们一块儿看下，专栏中用到的所有Java语言的语法。

Hello World

我们先来看一下，Java语言的Hello World代码如何编写。

在Java中，所有的代码都必须写在类里面，所以，我们定义一个HelloWorld类。main()函数是程序执行的入口。main()函数中调用了Java开发包JDK提供的打印函数System.out.println()来打印hello world字符串。除此之外，Java中有两种代码注释方式，第一种是“//注释…”双斜杠，表示后面的字符串都是注释，第二种是“/*注释…*/”，表示中间的内容都是注释。

```
/*hello world程序*/
public class HelloWorld {
    public static void main(String []args) {
        System.out.println("Hello World"); //打印Hello World
    }
}
```

基本数据类型

Java语言中的基本数据类型跟其他语言类似，主要有下面几种：

- 整型类型：byte（字节）、short（短整型）、int（整型）、long（长整型）
- 浮点类型：float（单精度浮点）、double（双精度浮点）
- 字符型：char
- 布尔型：boolean

如下，我们来定义一个基本类型变量：

```
int a = 6;
```

除此之外，为了方便我们使用，Java还提供了一些封装这些基本数据类型的类，这些类实现了一些常用的功能函数，可以直接拿来使用。常用的有下面几个类：

- Integer：对应封装了基本类型int；
- Long：对应封装了基本类型long；
- Float：对应封装了基本类型float；
- Double：对应封装了基本类型double；
- Boolean：对应封装了基本类型boolean；
- String：对应封装了字符串类型char[]。

如下，我们来定义一个Integer对象：

```
Integer oa = new Integer(6);
```

数组

Java中，我们使用[]来定义一个数组，如下所示：

```
int a[] = new int[10]; //定义了一个长度是10的int类型数组
```

在Java中，我们通过如下方式访问数组中的元素：

```
a[1] = 3; //将下标是1的数组元素赋值为3
System.out.println(a[2]); //打印下标是2的数组元素值
```

流程控制

流程控制语句跟其他语言类似，主要有下面几种。

- if-else语句，代码示例如下所示：

```
// 用法一
int a;
if (a > 1) {
  //执行代码块
} else {
  //执行代码块
}

// 用法二
int a;
if (a > 1) {
  //执行代码块
} else if (a == 1) {
  //执行代码块
} else {
  //执行代码块
}
```

- switch-case语句，代码示例如下所示：

```
int a;
switch (a) {
  case 1:
    //执行代码块
    break;
  case 2:
    //执行代码块
    break;
  default:
    //默认执行代码
}
```

- for、while循环，代码示例如下所示：

```
for (int i = 0; i < 10; ++i) {
  // 循环执行10次此代码块
}

int i = 0;
while (i < 10) {
  // 循环执行10次此代码块
}
```

- continue、break、return，代码示例如下所示：

```
for (int i = 0; i < 10; ++i) {
  if (i == 4) {
    continue; //跳过本次循环，不会打印出4这个值
  }
  System.out.println(i);
}

for (int i = 0; i < 10; ++i) {
  if (i == 4) {
    break; //提前终止循环，只会打印0、1、2、3
  }
  System.out.println(i);
}

public void func(int a) {
  if (a == 1) {
    return; //结束一个函数，从此处返回
  }
  System.out.println(a);
}
```

类、对象

Java语言使用关键词class来定义一个类，类中包含成员变量（也叫作属性）和方法（也叫作函数），其中有一种特殊的函数叫作构造函数，其命名比较固定，跟类名相同。除此之外，Java语言通过new关键词来创建一个类的对象，并且可以通过构造函数，初始化一些成员变量的值。代码示例如下所示：

```
public class Dog { // 定义了一个Dog类
  private int age; // 属性或者成员变量
  private int weight;

  public Dog(int age, int weight) { // 构造函数
    this.age = age;
    this.weight = weight;
  }

  public int getAge() { // 函数或者方法
    return age;
  }

  public int getWeigt() {
    return weight;
  }

  public void run() {
    // ...
  }
}

Dog dog1 = new Dog(2, 10);//通过new关键词创建了一个Dog对象dog1
int age = dog1.getAge();//调用dog1的getAge()方法
dog1.run();//调用dog1的run()方法
```

权限修饰符

在前面的代码示例中，我们多次用到private、public，它们跟protected一起，构成了Java语言的三个权限修饰符。权限修饰符可以修饰函数、成员变量。

- private修饰的函数或者成员变量，只能在类内部使用。
- protected修饰的函数或者成员变量，可以在类及其子类内使用。
- public修饰的函数或者成员变量，可以被任意访问。

除此之外，权限修饰符还可以修饰类，不过，专栏中所有的类定义都是public访问权限的，所以，我们可以不用去了解三个修饰符修饰类的区别。

对于权限修饰符的理解，我们可以参看下面的代码示例：

```
public class Dog {// public修饰类
  private int age; // private修饰属性，只能在类内部使用
  private int weight;

  public Dog(int age, int weight) {
    this.age = age;
    this.weight = weight;
  }

  public int getAge() { //public修饰的方法，任意代码都是可以调用
    return age;
  }

  public void run() {
    // ...
  }

}
```

继承

Java语言使用extends关键字来实现继承。被继承的类叫作父类，继承类叫作子类。子类继承父类的所有非private属性和方法。具体的代码示例如下所示：

```
public class Animal { // 父类
  protected int age;
  protected int weight;

  public Animal(int age, int weight) {
    this.age = age;
    this.weight = weight;
  }

  public int getAge() { // 函数或者方法
    return age;
  }

  public int getWeigt() {
    return weight;
  }

  public void run() {
    // ...
  }
```

```
  }
}

public class Dog extends Animal { // 子类
  public Dog(int age, int weight) { // 构造函数
    super(age, weight); //调用父类的构造函数
  }

  public void wangwang() {
    //...
  }
}

public class Cat extends Animal { //子类
  public Cat(int age, int weight) { // 构造函数
    super(age, weight); //调用父类的构造函数
  }

  public void miaomiao() {
    //...
  }
}

//使用举例
Dog dog = new Dog(2, 8);
dog.run();
dog.wangwang();
Cat cat = new Cat(1, 3);
cat.run();
cat.miaomiao();
```

接口

Java语言通过interface关键字来定义接口。接口中只能声明方法，不能包含实现，也不能定义属性。类通过implements关键字来实现接口中定义的方法。在专栏的第8讲中，我们会详细讲解接口，所以，这里我只简单介绍一下语法。具体的代码示例如下所示：

```
public interface Runnable {
  void run();
}

public class Dog implements Runnable {
  private int age; // 属性或者成员变量
  private int weight;

  public Dog(int age, int weight) { // 构造函数
    this.age = age;
    this.weight = weight;
  }

  public int getAge() { // 函数或者方法
    return age;
  }

  public int getWeigt() {
    return weight;
  }

  @Override
  public void run() { //实现接口中定义的run()方法
    // ...
  }
}
```

容器

Java提供了一些现成的容器。容器可以理解为一些工具类，底层封装了各种数据结构。比如ArrayList底层就是数组，LinkedList底层就是链表，HashMap底层就是散列表等。这些容器我们可以拿来直接使用，不用从零开始开发，大大提高了编码的效率。具体的代码示例如下所示：

```
public class DemoA {
  private ArrayList<User> users;

  public void addUser(User user) {
    users.add(user);
  }
}
```

异常处理

Java提供了异常这种出错处理机制。我们可以指直接使用JDK提供的现成的异常类，也可以自定义异常。在Java中，我们通

过关键字throw来抛出一个异常，通过throws声明函数抛出异常，通过try-catch-finally语句来捕获异常。代码示例如下所示：

```
public class UserNotFoundException extends Exception { // 自定义一个异常
  public UserNotFoundException() {
    super();
  }

  public UserNotFoundException(String message) {
    super(message);
  }

  public UserNotFoundException(String message, Throwable e) {
    super(message, e);
  }
}

public class UserService {
  private UserRepository userRepo;
  public UserService(UseRepository userRepo) {
    this.userRepo = userRepo;
  }

  public User getUserById(long userId) throws UserNotFoundException {
    User user = userRepo.findUserById(userId);
    if (user == null) { // throw用来抛出异常
      throw new UserNotFoundException();//代码从此处返回
    }
    return user;
  }
}

public class UserController {
  private UserService userService;
  public UserController(UserService userService) {
    this.userService = userService;
  }

  public User getUserById(long userId) {
    User user = null;
    try { //捕获异常
      user = userService.getUserById(userId);
    } catch (UserNotFoundException e) {
      System.out.println("User not found: " + userId);
    } finally { //不管异常会不会发生，finally包裹的语句块总会被执行
```

```
      System.out.println("I am always printed.");
    }
    return user;
  }
}
```

package包

Java通过pacakge关键字来分门别类地组织类，通过import关键字来引入类或者package。具体的代码示例如下所示：

```
/*class DemoA*/
package com.xzg.cd; // 包名com.xzg.cd

public class DemoA {
  //...
}

/*class DemoB*/
package com.xzg.alg;

import java.util.HashMap; // Java工具包JDK中的类
import java.util.Map;
import com.xzg.cd.DemoA;

public class DemoB {
  //...
}
```

总结

今天，我带你一块学习了专栏中用到的所有的Java基本语法。不过，我希望你不要纠结于专栏或者某某书籍到底是用什么编程语言来写的。语言层面的东西完全不会限制我的讲解和你的理解。这就像我们读小说一样，不管它是用英语写的，还是中文写的，故事都可以同样精彩。而且，多了解一些Java语法，对于你今后阅读Java语言编写的书籍或者文档，也很有帮助。

实际上，我之前在Google工作的时候，大家都不太在意自己熟悉的是哪种编程语言，很多同事都是“现学现卖”，什么项目适合用什么语言就现学什么语言。除此之外，Google在招聘的时候，也不限定候选人一定要熟悉哪种编程语言，也很少问跟语言特性相关的问题。因为他们觉得，编程语言只是一个工具，对于一个有一定学习能力的人，学习一门编程语言并不是件难事。

除此之外，对于专栏中的代码示例，你也可以用你熟悉语言重新实现一遍，我相信这也是件很有意义的事情，也更能加深你对内容的理解。

课堂讨论

不同的公司开发使用的编程语言可能不一样，比如阿里一般都是用Java，今日头条用Go、C++比较多。在招聘上，这些公司都倾向于招聘熟悉相应编程语言的同学，毕竟熟练掌握一门语言也是要花不少时间的，而且用熟悉的编程语言来开发，肯定会更得心应手，更不容易出错。今天课堂讨论的话题有两个：

1. 分享一下你学习一门编程语言的经历，从入门到熟练掌握，大约花了多久的时间？有什么好的学习编程语言的方法？
2. 在一个程序员的技术能力评价体系中，你觉得“熟练使用某种编程语言”所占的比重有多大？

欢迎在留言区写下你的想法，和同学一起交流和分享。如果有收获，也欢迎你把这篇文章分享给你的朋友。

精选留言

辣么大

Java用的时间最长，大概4-5年，不敢说自己“熟练”掌握了。最近反而觉得不懂的更多了。我没有抓入Java8不放，而是跟着Java的发展，开始学习Java11和Java13的新特性，紧跟语言的变化，并学习虚拟机和并发编程的相关知识。

我觉得熟练使用某种编程语言，在技术能力评价中占比起码50%。因为“熟练”是衡量一名程序员对一门语言掌握程度的重要指标。说明他\她不但会使用该语言，而且知道语言的细节，优缺点、适用场合等等。从入门到掌握、精通一门语言或者技能，是要花大功夫的，能看出一个人是否能把一件事（可能很枯燥）做到极致，有是否钻研的精神。这种能力是很多人不具备的。

国内招人还是很实际的，来了就能干活。学习能力是很虚的一个东西，嘴上说说没啥用。熟练掌握一门语言，才能看出你有学习能力。

2019-11-17 06:55

编程界的小学生

原谅我这篇文章三十秒就看完了，因为我是JAVA

1.用了多久我也不确定，但是学习方法是有的，首先看视频，资料，动手敲，晚上睡觉前在脑海里回顾一下学了什么，明天在动手敲一遍昨天学的内容，最后用自己的语言将其组织成一篇属于自己的文章。

2.熟练需要看成都，就比如很多人都说看源码感觉没用，看了就忘，也不知道能干嘛。我认为看源码首先能隐形锻炼你写代码的风格，学习里面的架构设计思想，且遇到奇葩问题你能知道怎么debug进去找问题，这些才是最主要的。我个人认为，如果没有看懂看清他里面的设计思想和核心源码，那我觉得你只是掌握了他的api，而不是熟悉。

2019-11-17 10:28

李小四

从第一次接触Java，到得心应手，大概花了两年时间。这个周期让我理解了学习的非线性。

大一开始学习C语言，学的似懂非懂，做了课程设计就放下了，发大二开始学Java，同样似懂非懂。大三开始接触Android开发，用到了Java，才发现自己Java知识不足，于是花时间重学了Java，过程中发现有些东西不理解，又穿插着把C需要的指针内存啃了几遍，大三结束的时候，Java才算熟练了，距离刚开始学习过去将近两年，中途无数次被打击，也放弃了很多次，因为每个字都认识，但看一次两次根本不理解，直到某一天你发出了一声恍然大悟的“哦～～～”，这种非线性的特点应该是很多人最终放弃的原因吧，一次次被打击，多次尝试没有正反馈，出于自我保护心理，说服自己放弃了。

2019-11-17 15:59

丁源(156*****518)

常人嗑语言，高手玩算法，大师谈模式，神人给定律。计算机，数学，哲学，神学。对抽象的理解越发深刻，构建高层次，高维度的模型就越稳定，由此意识形态泛化出的物质形态也会跟着越接近现实。

2019-11-22 00:33

摸爬滚打三十年

因为看老师的专栏太入迷，坐过了站

2019-11-29 08:55

极客不落

Day013 加餐一

勘误：第二种是“/注释…/”

应为：第二种是“/* 注释…*/”

2019-11-17 20:19

作者回复

嗯嗯 多谢指出

2019-11-18 12:12

Jason

大学开始主要了两年Java，然后毕业找到深度学习的岗位，主要用python和c++。

有java基础，上手就很快，学习python就跟着GitHub上一个python 100 days那个项目学了一个礼拜，但没看web相关的，主要是看机器学习相关的一些库怎么用。反正现在能用tensorflow调调参。

然后两周前又转回Java，感受就是语言层面的没设计模式，数据结构与算法重要，更没分析问题，解决问题的能力重要。

2019-11-19 00:09

小马哥

大学课程中学习了C，工作中自学并使用JAVA，主要用于web和大数据开发，JAVA不仅仅是一门语言，还是一个技术体系，包括了后来的很多技术框架，学习JAVA语言如果有其他语言基础是很快的，精通后面的一些常用框架就需要一些设计模式的积累。所以还是学习能力最重要：学习，操练，总结，分享，这个路线是我认为很快捷的学习方法。最后学习的东西越多，越容易融会贯通，后来使用Python做推荐系统，我们几个JAVA开发人员，基本是用一个小时过了一遍Python语法，就开工了

2019-11-17 13:59

被水淹没

高中学了vb，以为大多编程语言都差不多，基本的变量、控制语句都差不多
然后去学C，看到后面... 我擦这指针啥玩意？看来得补一下指针，然后搜了下.. 厚厚的一本指针教程... 然后就被打击到了
直到大学学了java，没指针，舒坦... 大学完就只会ssh，ssm，还好实习找得到工作，现在努力提升自己，先把争哥的这2套课程学了先 ^-^

2019-11-18 16:48

阿辉

Swift， Object-c，Java

2019-11-18 00:47

相逢是缘

非计算机专业，在学校就学习了C语言，在工作中也一直使用C。后来自己看了C++，JAVA。学习一门语言最开始就是一些基本语法和数据结构，了解了这两个就可以自己调试一些简单的demo。接下来就是你的程序不能只在控制台打印信息，要能对外输入和输出。这块一般是两部分:1、还是在自己的计算机内部，能从磁盘读取和保存数据。2、能够和别的计算机通信，也就是socket http等网络编程。再接下来就是一些每个编程语言自己特有的特性，需要好好琢磨和体会。深入一些就是编写的程序到计算机执行是个什么过程，对c来讲，程序如何被编译的，如何链接的，如何被系统装载运行的；对JAVA来讲，需要了解JVM。只有了解了这些，才能了解前辈大牛们的程序为什么那样去写，也才能解决后面遇到的一些深层次的问题。

2019-11-17 08:19

偶然~zZ

啥时候出一篇 python所有语法啊

2019-11-18 18:13

作者回复

可能暂时不会了

2019-11-19 10:07

Lyre

这篇有点混

2019-11-18 09:32

作者回复

这是加餐啊 大哥

2019-11-18 11:23

halweg

小争哥真的是在手把手扶着我们往过淌

2019-11-18 04:19

斜杠ing...

语言虽然只是个工具，但对于大多数人来说，几乎就用一种语言，还是精通后才能说语言是相通的。

2019-11-17 10:51

William

居然有加餐，666.

一篇文章涵盖java 语法，赞.

另外补充一下，关于权限限定符，
还有个default，支持的范围是 本包内可用，

protected 也支持本包内可使用.

编程语言确实不是最重要的，因为在企业中需要的是能够产生价值的软件，而非其他.

2019-11-17 10:45

稳

1、拿个人Python的经历，入门3天，熟练的话需要至少写一个项目，工作写了2年也不敢吹精通。我觉得学习一门语言，入门时一定要打牢基础，把基础语法耐心看仔细了，以后才会少踩坑，书籍专栏视频都可以。熟练的话一定要多练，主要时熟悉常用库和生态，精通就需要研究源码和不断的思考了。

2、虽然语言是工具，我觉得必须要有一门足够熟练的编程语言，这样整个人思维深度都不一样。从国内招聘来说，也只愿意要来了就能赚钱的，而不是花钱培养还可能跑路的。

2019-11-17 10:38

阿冰777

学过N门语言，基本触类旁通，熟练的只有java和js吧，技巧就是多练，多看，多想，多练就不说了，多看就是多看源码，多看别人是怎么实现的，多想就是琢磨为什么要这么写不那么写。

熟练掌握算是基本，只要你认真学，勤于思考，肯定是很快可以熟练掌握的，掌握了一门语言，也只是开始而已。

2019-12-06 23:33

日拱一卒

这篇看得毫无压力

2019-11-29 17:52

无名氏

这篇毫无压力

2019-11-26 21:55