我们先来简单地看一下什么是类，对象，方法和实例变量。

****对象****- 对象具有状态和行为。例如：狗有状态 - 颜色，名称，繁殖以及行为，摇头晃脑，吠叫，吃。对象是类的实例。

****类**** - 一个类可以被定义为描述行为的模板/蓝色印花/指出其类型支持的对象。

****方法**** - 一种方法，基本上是一个行为。一个类可以包含许多方法。它是在将逻辑写入时，数据操纵和所有的动作被执行的方法。

****实例变量**** - 每个对象都有其独特的实例变量。一个对象的状态是由分配给这些实例变量的值来创建。

## **1、第一个Java程序：**

看看打印单词 Hello World 的简单代码。

|  |
| --- |
| public class MyFirstJavaProgram {  /\* This is my first java program.  \* This will print 'Hello World' as the output  \*/  public static void main(String []args) {  System.out.println("Hello World"); // prints Hello World  }} |

看下面是如何保存文件，编译并运行该程序。请按照以下步骤进行：

打开记事本，并添加如上面的代码。

将文件另存为：MyFirstJavaProgram.java。

打开命令提示符窗口，然后去进入保存类的目录。假设它是C：。

输入'javac MyFirstJavaProgram.java“，然后按回车来编译代码。如果代码中没有错误，在命令提示符下将到下一行（假设：路径变量设置）。

现在，输入“java MyFirstJavaProgram'来运行程序。

将可以看到 ' Hello World ' 印在窗口上。

C : > javac MyFirstJavaProgram.java

C : > java MyFirstJavaProgram Hello World

## **2、基本语法：**

关于Java程序，要记住以下几点。

****大小写敏感性**** - Java是大小写敏感的，这意味着标识符Hello 和 hello会有不同的含义在Java中。

****类名称**** - 对于所有的类名首字母应该大写。  
如果需要用几个单词来构成类的名称，每个内部单词的第一个字母应该大写。  
如类名 MyFirstJavaClass

****方法名称****- 所有的方法名应该以小写字母开头。  
如果几个单词来构成方法的名称，每个内部单词的第一个字母应该大写。  
例如 public void myMethodName()

****程序文件名**** - 程序文件的名称应的类的名称完全匹配。  
保存文件时，应该使用类名（请记住Java是大小写敏感）并添加 '.java'的名称的末尾（如果该文件名和类名不符合程序将无法编译）保存。  
例如: 假设'MyFirstJavaProgram“是类名。那么该文件应保存为 'MyFirstJavaProgram.java'

****public static void main(String args[])****- Java程序处理从main（）方法开始，这是每一个Java程序的强制性部分入口..

## **3、Java标识符：**

所有的Java组件都需要名字。用于类，变量和方法的名字叫做标识符。

在Java中，有几点要记住的标识符。它们如下：

所有的标识符应该以一个字母（A至Z或a到z），货币字符（$）或下划线（\_）。

之后的第一个字符的标识符可以具有字符的任意组合。

关键字不能被用作标识符。

最重要是标识符是区分大小写的。

合法标识符的例子: age, $salary, \_value, \_\_1\_value

非法标识符的例子s: 123abc, -salary

## **4、Java修饰符：**

像其他的语言，它是可以修改的类，方法等，通过使用修饰符。有两类修饰符：

****访问修饰符****: default, public , protected, private

****非访问修饰符****: final, abstract, strictfp

关于修饰符在下一节会给出更多细节。

## **5、Java变量：**

我们会看到以下变量类型在Java中：

* 局部变量
* 类变量（静态变量）
* 实例变量（非静态变量）

## **6、Java数组：**

数组是存储相同类型的多个变量的对象。然而一个数组本身是在堆中的对象。我们会学习如何声明，构造和初始化在即将到来的章节。

## **7、Java 枚举:**

枚举在Java 5.0中进行了介绍。枚举一个变量限于只有几个预定义的值之一。在此枚举列表的值称为枚举。

随着使用枚举，可以减少错误代码的数量。

例如，如果考虑一个新鲜果汁店的应用程序，这将有可能限制玻璃尺寸小，中，大。这将确保它不会允许任何人以任何尺寸比小型，中型或大型的订单。

例子:

class FreshJuice {

enum FreshJuiceSize{ SMALL, MEDIUM, LARGE }

FreshJuiceSize size;}

public class FreshJuiceTest {

public static void main(String args[]){

FreshJuice juice = new FreshJuice();

juice.size = FreshJuice. FreshJuiceSize.MEDIUM ;

System.out.println("Size: " + juice.size);

}}

上面的例子会产生以下结果：

Size: MEDIUM

****注：****枚举可以声明为自己或一个类的内部。方法，变量，构造函数可以枚举内定义为好。

## **8、Java 关键字:**

下面列出了Java中的保留字。这些保留字可以不被用作常量或变量，或任何其他的标识符名称。

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | assert | boolean | break |
| byte | case | catch | char |
| class | const | continue | default |
| do | double | else | enum |
| extends | final | finally | float |
| for | goto | if | implements |
| import | instanceof | int | interface |
| long | native | new | package |
| private | protected | public | return |
| short | static | strictfp | super |
| switch | synchronized | this | throw |
| throws | transient | try | void |
| volatile | while |  |  |

## **9、Java的注释**

Java支持单行和多行注释非常类似于C和C + +。任何注释内所有可用的字符是由Java编译器忽略。

public class MyFirstJavaProgram{

/\* This is my first java program.

\* This will print 'Hello World' as the output

\* This is an example of multi-line comments.

\*/

public static void main(String []args){

// This is an example of single line comment

/\* This is also an example of single line comment. \*/

System.out.println("Hello World");

}}

## **10、使用空行：**

只含有空格，可能带有注释行，被称为一个空行，Java完全忽略它。

## **11、继承：**

在Java中，类可以衍生自类。基本上，如果需要创建一个新的类，在这里已经有一些需要的代码的类，然后就可以从已经存在的代码中派生新类。

这个概念允许重用现有类的字段和方法，而无需重写代码在一个新的类。在这种情况下现有的类称为父类和派生类称为子类。

## **12、接口：**

在Java语言中，接口可以被定义为关于如何相互通信的对象之间的联系。接口起到了至关重要的作用，当涉及到继承的概念。

接口定义的方法，一个派生类（子类）应该使用。但该方法的实现完全取决于子类。