在本章课程中我们将学习 Java 基本语法里的关键字、标识符、注释、变量和常量等的相关知识。

**2.1 Java 程序的构成**

一个Java程序可以认为是一系列对象的集合，而这些对象通过调用彼此的方法来协同工作。

* 对象(object)：对象是类的一个实例，有状态和行为。例如，一条狗是一个对象，它的状态有：颜色、名字、品种；行为有：摇尾巴、叫、吃等。
* 类(class)：类是一个模板，它描述一类对象的行为和状态。
* 方法(method)：方法就是行为，一个类可以有很多方法。逻辑运算、数据修改以及所有动作都是在方法中完成的。
* 实例变量：每个对象都有独特的实例变量，对象的状态由这些实例变量的值决定。
* 还记得上一节课的 Hello World 吗？这一节课，我们还是先来看一看这一段代码。

public class HelloWorld  
{  
 public static void main(String[] args)  
 {  
 System.out.println("Hello World!");  
 }  
}

在这里的 Hello World 中只涉及了类、方法和语句。该 Java 程序定义了一个类 HelloWorld，该类的名字与 .java 文件的名字相同,其中的 main 方法是程序的入口，而“打印”这个功能是由一个语句实现，即：System.out.println("Hello World!");。

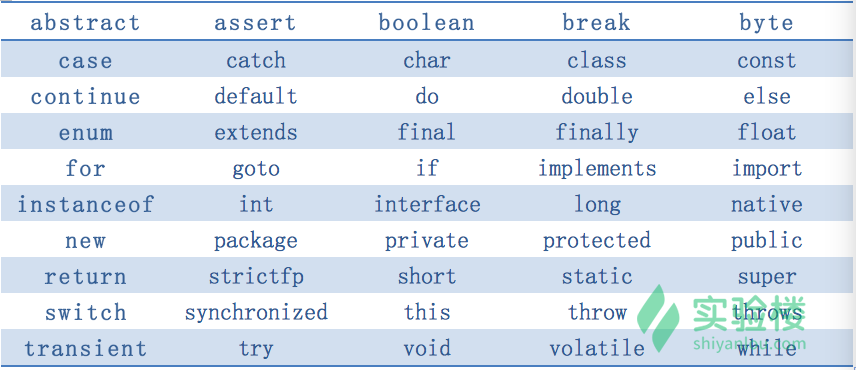
下面两点有关 Java 的书写方式：

* Java 中的语句要以;结尾 。
* 用花括号{}来整合语句，形成程序块。通过程序块，我们可以知道程序的不同部分的范围，比如类从哪里开始，到哪里结束。

**2.2 Java 关键字**

Java 的关键字对 java 的编译器有特殊的意义，他们用来表示一种数据类型，或者表示程序的结构等，关键字不能用作变量名、方法名、类名、包名。

Java 关键字有如下表所列，目前共有50个Java关键字，其中，"const"和"goto"这两个关键字在Java语言中并没有具体含义。同学们先有个印象，具体含义我们将在后续的内容中详细讲解：



**2.3 Java 标识符**

**Java 语言中，类、变量、常量、方法都需要名字，我们统统称之为 Java 标识符.**

标识符是用来给类、对象、方法、变量、接口和自定义数据类型命名的。

关于 Java 标识符，有几点需要注意的：

* Java 标识符由数字，字母A-Z或者a-z和下划线\_，美元符号$组成。
* 所有的标识符都应该以字母A-Z或者a-z,美元符$、或者下划线\_开始，**首位不能是数字**。
* 关键字不能用作标识符。
* 在 Java 中是区分大小写的。

下面的标识符是合法的：

shiYanlou, Shi\_Yanlou, $money, \_Java\_learning, OK, \_3th, \_3\_

下面的标识符是非法的：

#shiyan，25years，class，&name，if

在 Java 中，还有一些约定俗成的命名规则，希望同学们在写代码的时候都能遵循这些规则：

1. **类和接口名。每个字的首字母大写**，含有大小写。例如，MyClass，HelloWorld，Time 等。
2. 方法名。**首字符小写，其余的首字母大写**，含大小写。尽量少用下划线。例如，myName，setTime 等。这种**命名方法叫做驼峰式命名**。
3. 常量名。**基本数据类型的常量名使用全部大写字母，字与字之间用下划线分隔**。对象常量可大小混写。例如，SIZE\_NAME。
4. 变量名。可大小写混写，首字符小写，字间分隔符用字的首字母大写。不用下划线，少用美元符号。给变量命名是尽量做到见名知意。
5. 命名过程中尽量做到见名知意，方便后期查看和修改代码，也方便其他人员的阅读。

**2.4 变量**

在程序中存在大量的数据来代表程序的状态，其中有些数据在程序的运行过程中值会发生改变，这些数据在程序中被叫做变量。

变量代表程序的状态。程序通过改变变量的值来改变整个程序的状态。为了方便地引用变量的值，在程序中需要为变量设定一个名称，这就是变量名。

由于Java语言是一种强类型的语言，所以变量在使用以前必须首先声明，在程序中声明变量的语法格式如下：

数据类型 变量名称；

例如：

int x = 1;

在该语法格式中，数据类型可以是Java语言中任意的类型，如int，下面会介绍数据类型的知识。变量名称是该变量的标识符，需要符合标识符的命名规则，数据类型和变量名称之间使用空格进行间隔，空格的个数不限，但是至少需要1个。语句使用；作为结束。

变量(variable)占据一定的内存空间。不同类型的变量占据不同的大小。Java 中的变量类型如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 数据类型 | 默认值 | 存储格式 | 数据范围 |
| short | 0 | 2个字节 | -32,768到32767 |
| int | 0 | 4个字节 | -2,147,483,648到2,147,483,647 |
| byte | 0 | 1个字节 | -128到127 |
| char | 空 | 2个字节 | Unicode的字符范围：’\u0000’（即为0）到’\uffff’（即为65,535） |
| long | 0L或0l | 8个字节 | -9,223,372,036,854,775,808到9,223,372,036, 854,775,807 |
| float | 0.0F或0.0f | 4个字节 | 32位IEEEE 754单精度范围 |
| double | 0.0或0.0D(d) | 8个字节 | 64位IEEE 754双精度范围 |
| boolean | false | 1位 | true(1)或false(0) |

在 Java 中，变量需要先声明(declare)才能使用。在声明中，说明变量的类型，赋予变量以特别名字，以便在后面的程序中调用它。你可以在程序中的任意位置声明变量。下面我们新建一个Test.java文件， 比如:

public class Test  
 {  
 public static void main(String[] args)  
 {  
 System.out.println("Define a variable a is ");  
 int a; //声明变量a  
 a = 5;  
 System.out.println(a); // 打印一个整数a  
 }  
 }

上面 a 是变量名。可以在声明变量的同时，给变量赋值，比如 int a = 5;。

Java中主要有如下几种类型的变量：

* 局部变量
* 类变量（静态变量）
* 成员变量（非静态变量）

变量的概念实际上来自于面向过程的编程语言。在Java中，所谓的变量实际上是基本类型 (premitive type)。我们将在类的讲解中更多深入。

**2.6 注释**

在2.4节的程序还可以看到，Java 中，可用//引领注释。

说到注释，在我们写代码的时候，为了方便理解和阅读，往往会在相关的代码附近添加说明文字，也就是我们的注释。注释是为了防止编译器在编译的时候将我们的注释也一并编译了，导致程序出错，所以在编译的时候，编译器会自动跳过我们编写的注释。

一般在 Java 里有三种注释：

* 行注释//：只注释一行
* 段注释/\*...\*/：注释若干行
* 文档注释/\*\*...\*/：注释若干行，**并写入 javadoc文档**

**2.7 自动类型转换和强制类型转换**

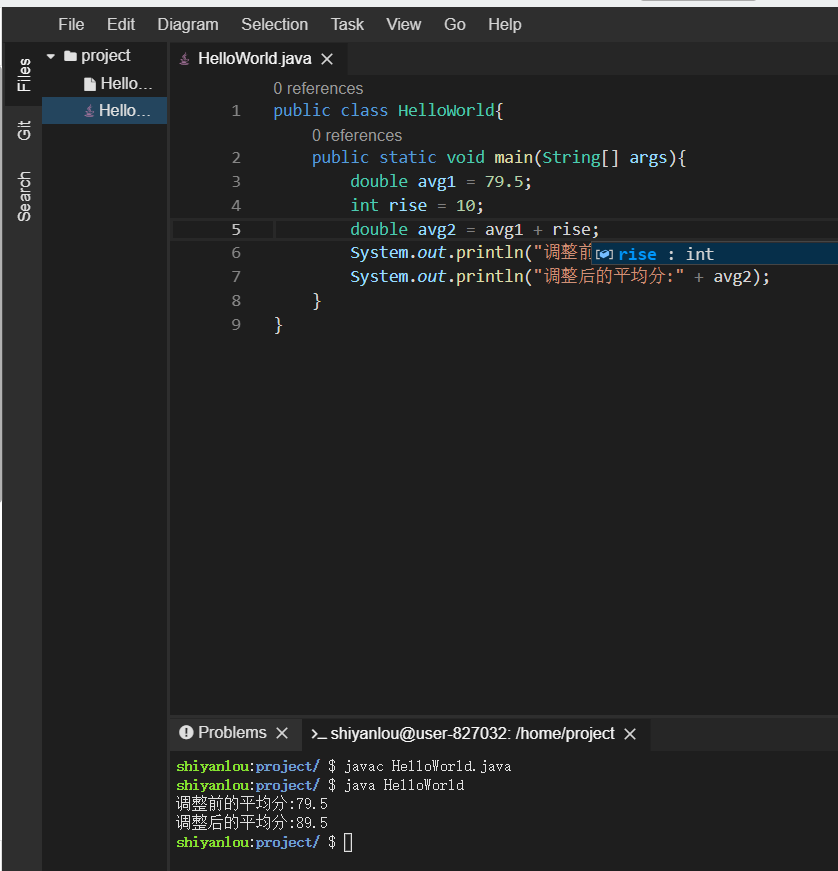
小伙伴们，我们在/home/project 下建立一个HelloWorld.java，编写并运行如下的程序段，在控制台输出的结果会是怎样的呢？

public class HelloWorld{  
 public static void main(String[] args) {  
 double avg1=79.5;  
 int rise=10;  
 double avg2=avg1+rise;  
 System.out.println("调整前的平均分："+avg1);  
 System.out.println("调整后的平均分："+avg2);  
 }  
}

提示：打开terminal，输入

javac HelloWorld.java  
java HelloWorld

就可以看到输出结果了！



在上面的程序中，我们定义了三个变量：avg1、rise、avg2，其中avg1是double类型的变量，而rise是int类型的变量，两种类型不一样的变量加在了一起变成了什么类型了呢？没错，在运算过程中rise被自动转换成了double类型的变量。

在Java程序中，不同的数据类型有些时候需要进行相互转换。数据类型转换就分为了自动类型转换和强制类型转换

自动类型转换是在程序执行过程中，不需要我们去特殊声明或者操作，变量由于需要而自动转换成了合适的数据类型。

自动类型转换需要满足下面的两个条件：

* 目标类型与原类型兼容
* 目标类型的字节数大于或等于原类型字节数

基本数据类型中，由于**布尔类型其本身所代表的特殊含义，不能与其他基本类型进行类型的转换（既不能进行自动类型的提升，也不能强制类型转换）， 否则，将编译出错。**

参考资料：

[Java基础语法](http://www.runoob.com/java/java-basic-syntax.html)

[Java常量和变量](http://www.cnblogs.com/langtianya/p/3868135.html)