# Java基础介绍

## 简介

* Java是美国 sun 公司（Stanford University Network）在1995年推出的一门计算机高级编程语言。
* Java 早期称为Oak(橡树)，后期改名为Java。
* Java 之父**：**詹姆斯·高斯林(James Gosling)。
* 2009年 sun公司被Oracle公司收购。

## 2. 为什么使用Java

* 世界上最流行的编程语言之一，在国内使用最为广泛的编程语言。
* 可移植性、安全可靠、性能较好。
* 开发社区最完善，功能最丰富。

## 3. Java技术体系

|  |  |
| --- | --- |
| **技术体系** | **说明** |
| Java SE(Java Standard Edition):标准版 | Java技术的核心和基础 |
| Java EE(Java Enterprise Edition):企业版 | 企业级应用开发的一套解决方案 |
| Java ME(Java Micro Edition):小型版 | 针对移动设备应用的解决方案 |

## 4. 如何使用Java

Java语言的产品是 **JDK（Java Development Kit ：Java开发者工具包) ，**必须安装JDK才能使用Java语言

**2014年JDK(8.0、LTS) yyds（常用于商业）LTS(long-term support):长期支持版**

**2021年9月14日JDK(17.0、LTS) （教学环境使用）**

## 5. 如何获取JDK

**去Oracle官网下载JDK** <https://www.oracle.com/java/technologies/downloads/>

## 6. 如何验证安装成功JDK?

按下 Win + R ，在运行输入框中输入 cmd，敲回车。

在命令行 窗口中分别输入javac –version 及 java –version看版本提示，

如果版本提示与自己安装的版本号一致，则代表JDK环境搭建成功

## 7. javac（编译工具）和Java（执行工具）介绍

我们写好的Java程序都是高级语言，计算机底层是硬件不能识别这些语言。

必须先通过javac编译工具进行翻译，然后再通过java执行工具执行才可以驱动机器干活

### 7.1 Java程序的执行原理是什么样的？

不管是什么样的高级编程语言，最终都是翻译成计算机底层可以识别的机器语言。

### 7.2 机器语言是由什么组成的啊？

0和1

### 7.3 JDK有哪些组成啊？

* JVM（Java Virtual Machine）虚拟机：真正运行Java程序的地方。
* 核心类库：Java自己写好的一些程序,给咱们的程序调用的
* 开发工具：javac、java、…

## 8. Java开发中常用命令行窗口命令

|  |  |
| --- | --- |
| **常用命令** | **作用** |
| 盘符： | 切换到某个盘下：D：， C: |
| dir | 查看当前路径下的文件信息 |
| cd | 进入单级目录: cd itheima  进入多级目录: cd D:\itheima\JavaSE\第一天  回退到上一级目录：cd ..  回退到盘符根目录：cd \ |
| cls | 清屏 |

## 9. JDK安装后Path和Java\_home环境变量

注：(较新版本的JDK会自动配置PATH环境变量，较老的JDK版本则不会)

**Path环境变量**用于记住程序路径，方便在命令行窗口（cmd）的任意目录启动程序

**Path环境变量的原理：**当我们在Path中配置某个程序路径后，启动命令行窗口启动程时，是如何去找该程序的

**JAVA\_HOME：**告诉操作系统JDK安装在了哪个位置（将来其他技术要通过这个环境变量找JDK）

JAVA\_HOME D:\soft\java\jdk-17.0.1

推荐：Path %JAVA\_HOME%\bin

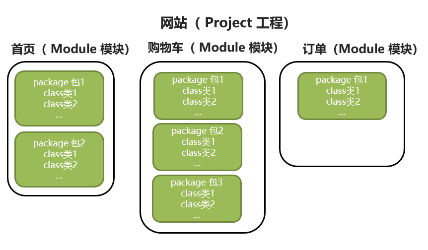
不推荐：Path D:\soft\java\jdk-17\bin

## 10. IDEA开发工具的使用

### 10.1 IDEA的下载和安装

下载链接：<https://www.jetbrains.com/idea/>

### 10.2 IDEA管理Java程序的结构

project（项目、工程）

module（模块）

package（包）

class（类）

### 10.3 使用idea开发第一个Java程序的步骤project -> module -> package –> class

1. 创建工程 new Project（空工程）（project中可以创建多个module）
2. 创建模块 new Module（module中可以创建多个package）
3. 创建包 new Package（package中可以创建多个class）
4. 创建类
5. 编写代码、并启动

### 10.4 IDEA中的java程序是自动编译和执行的，那编译后的class文件在哪里?

储存在工程路径下的 一个out 文件夹里

### 10.5 IDEA常用快捷键

|  |  |
| --- | --- |
| **IDEA常用快捷键** | **功能效果** |
| main/psvm、sout、… | 快速键入相关代码 |
| Ctrl + D | 复制当前行数据到下一行 |
| Ctrl + Y | 删除所在行，建议用Ctrl + X |
| Ctrl + ALT + L | 格式化代码 |
| ALT + SHIFT + ↑ , ALT + SHIFT + ↓ | 上下移动当前代码 |
| Ctrl + / , Ctrl + Shift + / | 对代码进行注释 |

# 入门程序HelloWord

开发 Java 程序，需要三个步骤：

编写代码 HelloWorld.java （源代码文件）

编译代码（使用javac编译） HelloWorld.class （字节码文件）

运行代码

## 使用记事本编写HelloWord

注：文件名称必须与代码中的类名称一致。保存文件：ctrl + s



控制台输入javac HelloWord.java编译文件得到文件HellowWord.class

输入java HelloWord运行文件（注意不要加入class后缀，行业规范）



## 常见错误

* Windows的文件扩展名没有勾选
* 代码写了，但是忘记保存了
* 文件名和类名不一致。
* 大小写错误，单词拼写错误，存在中文符号，找不到main方法。
* 括号不配对。
* 编译或执行工具使用不当。

# 三. JAVA基础语法

## 注释（注释不影响程序的执行）

**单行注释**

// 注释内容，只能写一行

**多行注释**

/\*

注释内容1

注释内容2

\*/

**文档注释：文档注释的内容是可以提取到一个程序说明文档中去的**

/\*\*

注释内容

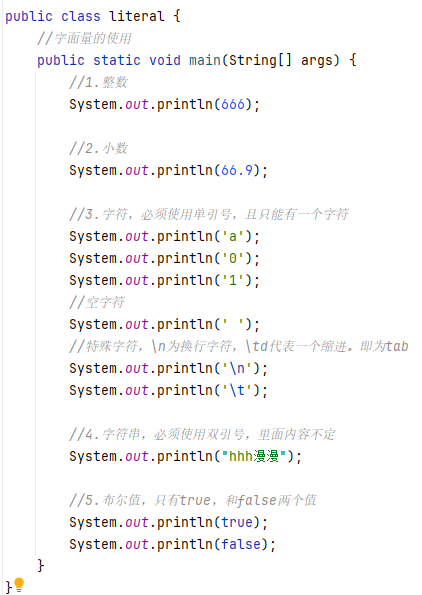
注释内容

\*/

|  |  |
| --- | --- |
| **IDEA快捷键进行注释** | **功能效果** |
| Ctrl + / | 单行注释（对当前行进行注释） |
| Ctrl + Shift + / | 对选中的代码进行多行注释。 |

## 字面量（告诉程序员数据在程序中的书写格式）

|  |  |  |  |
| --- | --- | --- | --- |
| **常用数据** | **生活中的写法** | **程序中的写法** | **说明** |
| 整数 | 666，-88 | 666，-88 | 写法一致 |
| 小数 | 13.14，-5.21 | 13.14，-5.21 | 写法一致 |
| 字符**（单引号）** | A, 0, 我 | ‘A’，‘0’， ‘我’ | 程序中必须使用单引号，有且仅能一个字符 |
| 字符串**（双引号）** | 黑马程序员 | “HelloWorld”，“黑马程序员” | 程序中必须使用双引号，内容可有可无 |
| 布尔值 | 真、假 | true 、false | 只有两个值：true：代表真，false：代表假 |
| 空值 |  | 值是：null | 一个特殊的值，空值 |

**几个常见的特殊值的书写格式：**

**true、 布尔类型：成功**

**false、布尔类型：失败**

**null、空值**

**\n、特殊字符：换行字符**

**\td 特殊字符：缩进字符**

## 变量（变量是用来记住程序要处理的数据的）

定义格式： 数据类型 变量名称 = (赋值) 数据

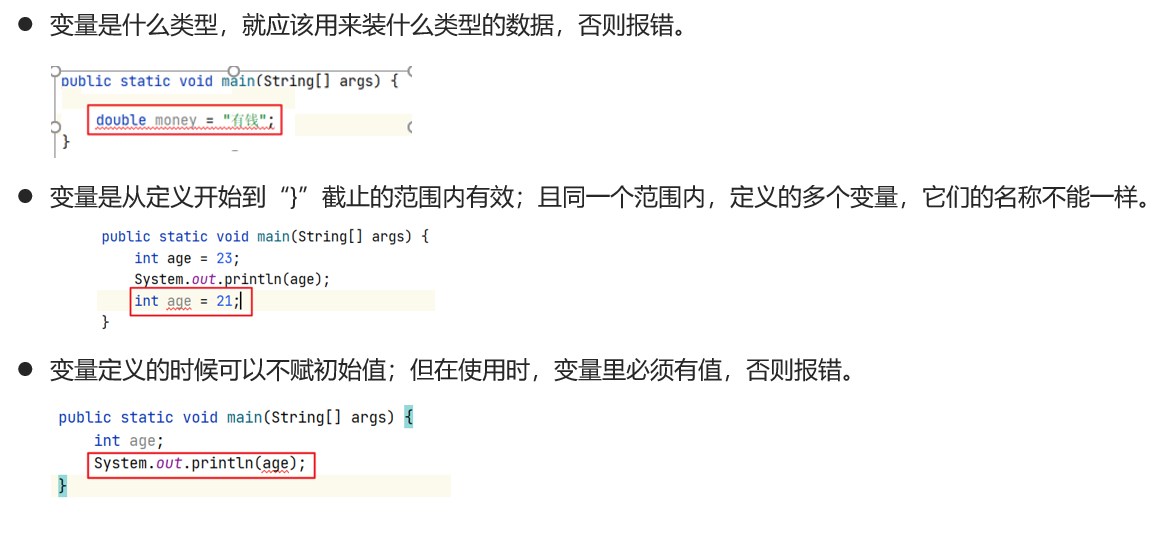
int age = 18;

数据类型：限制盒子中只能存储某种数据形式: 例如：int（整数类型）、 double(小数类型)

变量名称：首字母建议小写，有意义。（驼峰命名法）

使用变量记要处理的数据，编写的代码更灵活，管理代码更方便。

**注意事项：**



## 关键字

Java语言自己用到的一些词，有特殊作用的，我们称之为关键字

关键字是java用了的，我们就不能用来为做为：类名、变量名，否则会报错

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** | **strictfp** | **short** | **static** | **super** |
| **switch** | **synchronized** | **this** | **throw** | **throws** |
| **transient** | **try** | **void** | **volatile** | **while** |

## 标识符

标志符就是名字，我们写程序时会起一些名字，如类名、变量名等等都是标识符。

**要求：**

**基本组成：**由数字、字母、下划线(\_)和美元符($)等组成（驼峰命名法）

**强制要求：**不能以数字开头、不能用关键字做为名字、且是区分大小写的