

一. Java 基础介绍

1. 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/>

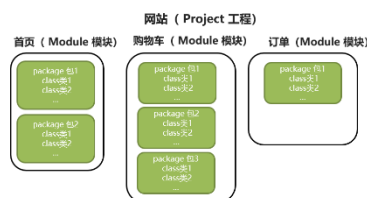
10.2 IDEA 管理 Java 程序的结构

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

- 创建工程 new Project (空工程) (project 中可以创建多个 module)
- 创建模块 new Module (module 中可以创建多个 package)
- 创建包 new Package (package 中可以创建多个 class)
- 创建类
- 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

二. 入门程序 HelloWorld

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

1. 使用记事本编写 HelloWorld

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 HelloWorld.java 编译文件得到文件 HelloWorld.class
输入 java HelloWorld 运行文件（注意不要加入 class 后缀，行业规范）

HellWorld程序介绍

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

class: 定义一个类，后面跟上的HelloWorld是类名称

程序执行时的入口点，main方法也称之为主方法

打印语句，使程序在控制台打印双引号所包裹的内容

2. 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

1. 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。

2. 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

```
public class literal {  
    //字面量的使用  
    public static void main(String[] args) {  
        //1. 整数  
        System.out.println(666);  
  
        //2. 小数  
        System.out.println(66.9);  
  
        //3. 字符，必须使用单引号，且只能有一个字符  
        System.out.println('a');  
        System.out.println('0');  
        System.out.println('1');  
        //空字符  
        System.out.println(' ');  
        //特殊字符，\n为换行字符，\td代表一个缩进。即为tab  
        System.out.println('\n');  
        System.out.println('\t');  
  
        //4. 字符串，必须使用双引号，里面内容不定  
        System.out.println("hhh嘿嘿");  
  
        //5. 布尔值，只有true，和false两个值  
        System.out.println(true);  
        System.out.println(false);  
    }  
}
```

3. 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：

- 变量是什么类型，就应该用来装什么类型的数据，否则报错。

```
public static void main(String[] args) {  
    double money = "有钱";  
}
```

- 变量是从定义开始到 “}” 截止的范围内有效；且同一个范围内，定义的多变量，它们的名称不能一样。

```
public static void main(String[] args) {  
    int age = 23;  
    System.out.println(age);  
    int age = 21;  
}
```

- 变量定义的时候可以赋初始值；但在使用时，变量里必须有值，否则报错。

```
public static void main(String[] args) {  
    int age;  
    System.out.println(age);  
}
```

4. 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

5. 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