

千锋Java学院出品

Java概述与环境搭建

Java Platform Standard Edition

课程目标

CONTENTS

ITEMS **1** 什么是程序

ITEMS **2** Java的历史与特点

ITEMS **3** Java的运行机制

ITEMS **4** Java的环境搭建

ITEMS **5** 第一个应用程序

ITEMS **6** Java的语言规范

什么是程序



即时聊天



路况、导航



便捷支付



美颜、修图

- 程序是为了**模拟现实世界**，**解决现实问题**而使用计算机语言编写的一系列有序的指令集合。

Java的历史

- Sun Microsystems于1995年推出的面向对象的程序设计语言，共同创始人的詹姆斯·高斯林 (James Gosling)被誉为“Java之父”。
- 1996年发布JDK 1.0
- 1998年发布JDK 1.2
 - 平台名称: J2SE(Standard Edition)、J2EE(Enterprise Edition)、J2ME(Micro Edition)
- 2004年发布JDK 1.5 版本更名: Java 5.0
 - 平台名称: Java SE、Java EE、Java ME
- 2006年发布Java 6.0
- 2009年被Oracle收购
- 2011年由Oracle发布Java 7.0
- 2014年由Oracle发布Java 8.0
- 2017年由Oracle发布Java 9.0
-
- 2019年由Oracle发布Java 14.0



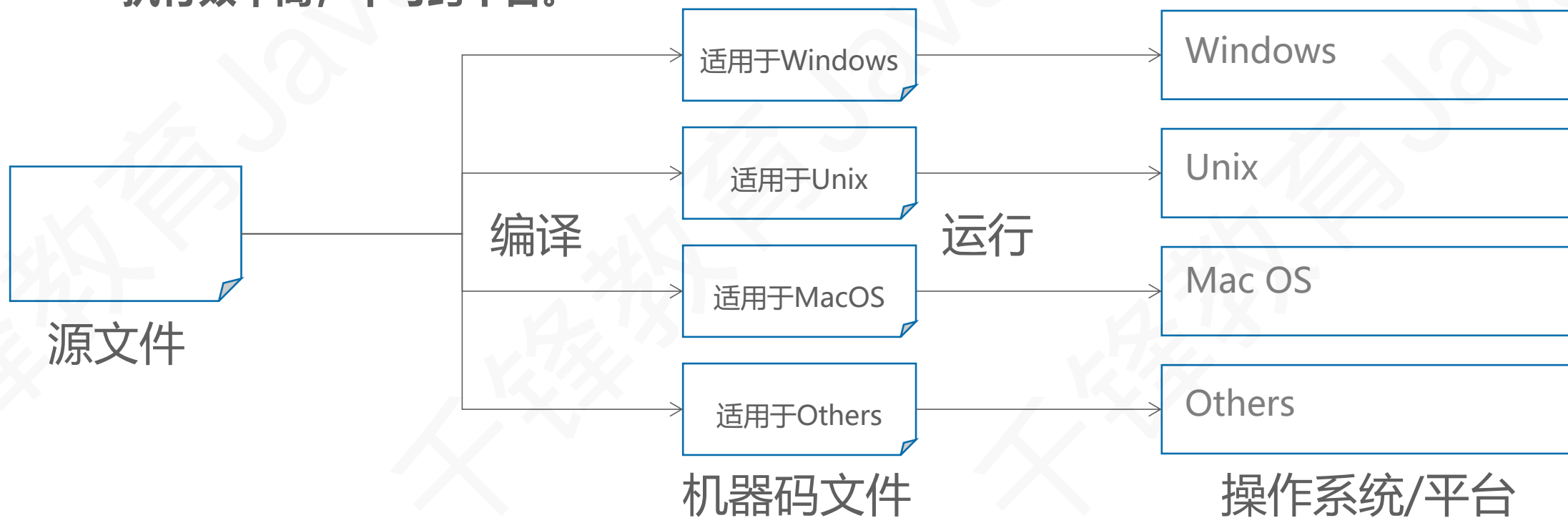
Java的语言特点

- 面向对象（贴近人类思维模式，模拟现实世界，解决现实问题）。
- 简单性（自动内存管理机制、不易造成内存溢出；简化流程处理、语义清晰）
- 跨平台（操作系统、服务器等）。

计算机的执行机制 (1)

- 编译执行:

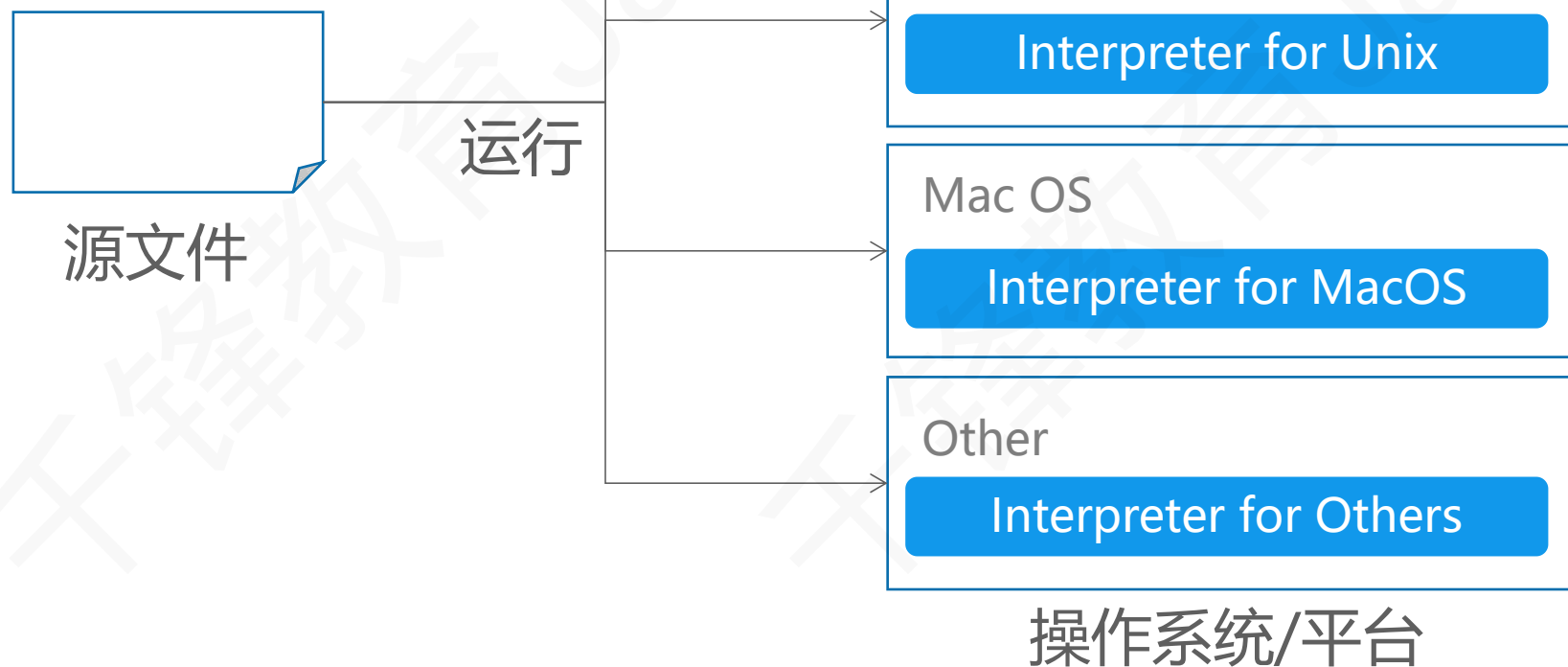
- 将源文件编译成平台相关的机器码文件, 一次编译, 多次执行。
- 执行效率高, 不可跨平台。



计算机的执行机制 (2)

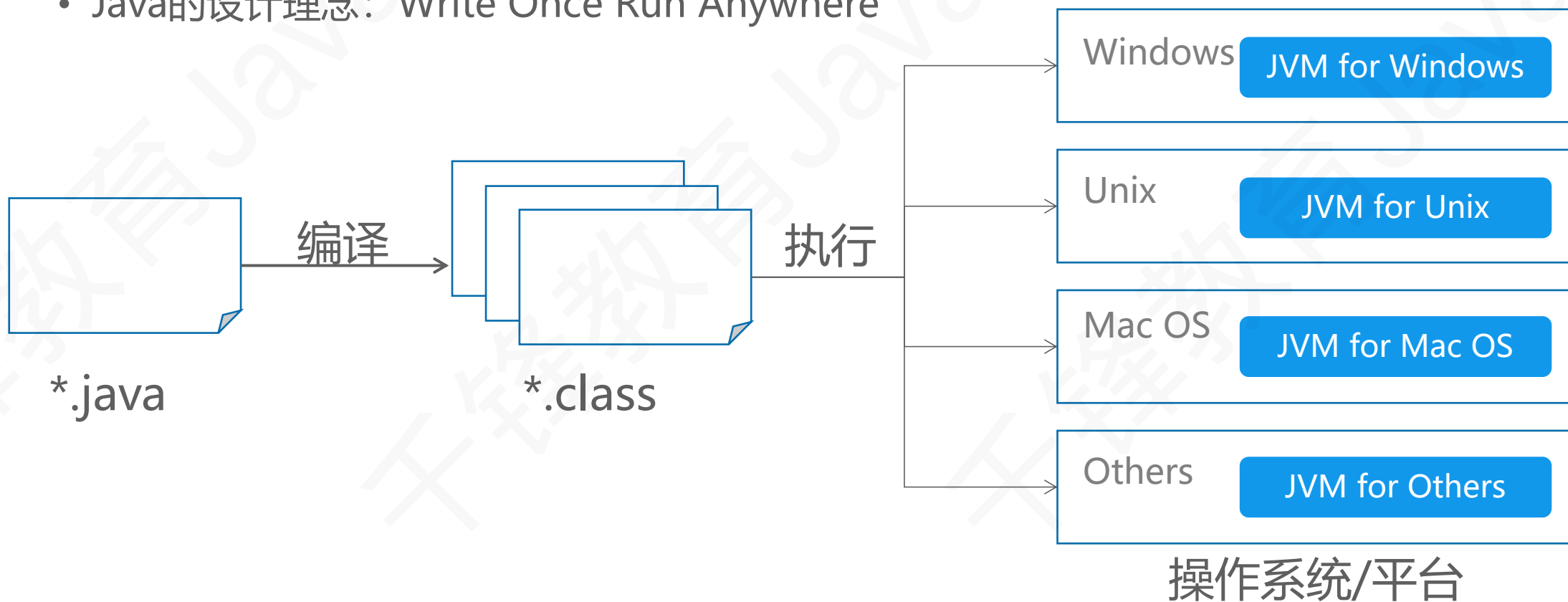
- 解释执行:

- 将源文件交给不同的平台独有的解释器。
- 执行效率低, 可以跨平台。



Java的执行机制

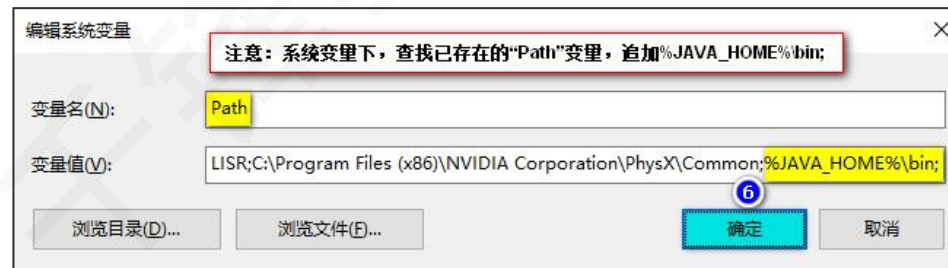
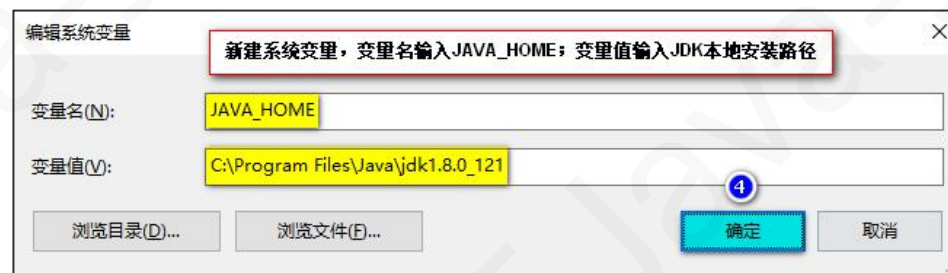
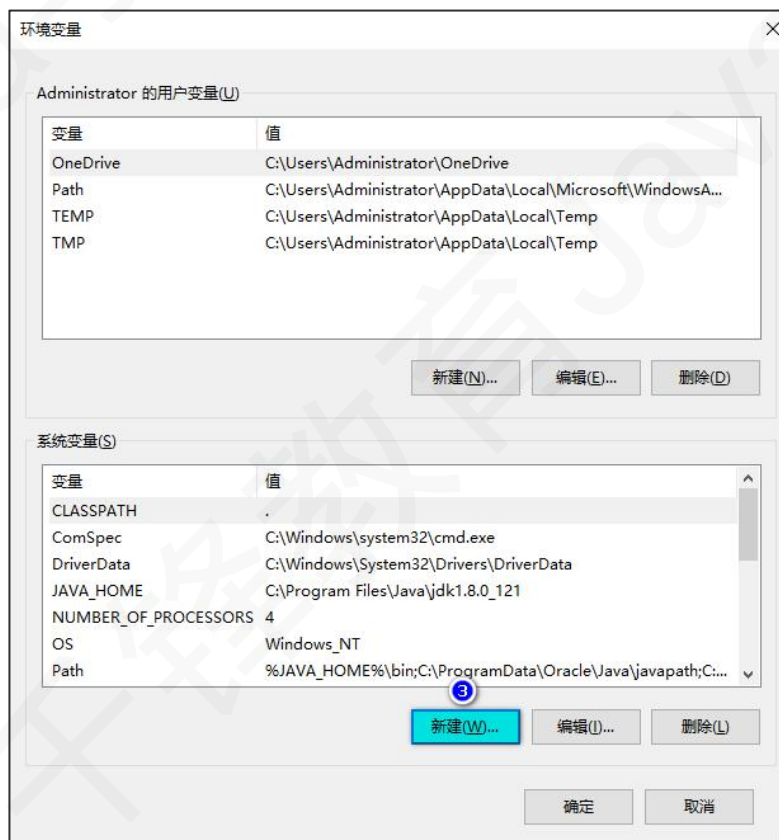
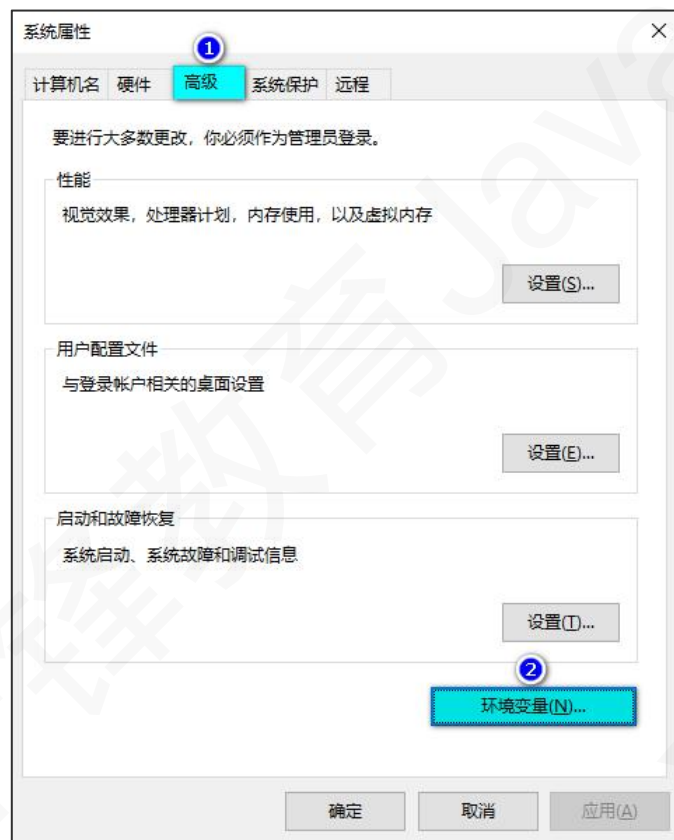
- 先编译、再解释：
 - 将源文件编译成字节码文件（平台中立文件.class），再将字节码文件进行解释执行。
 - Java的设计理念：Write Once Run Anywhere



- **JVM (Java Virtual Machine) 虚拟机:**
 - 使用软件在不同操作系统中, 模拟相同的环境。
- **JRE (Java Runtime Environment) 运行环境:**
 - 包含JVM和解释器, 完整的Java运行环境。
- **JDK (Java Development Kit) 开发环境:**
 - 包含JRE + 类库 + 开发工具包 (编译器+调试工具) 。

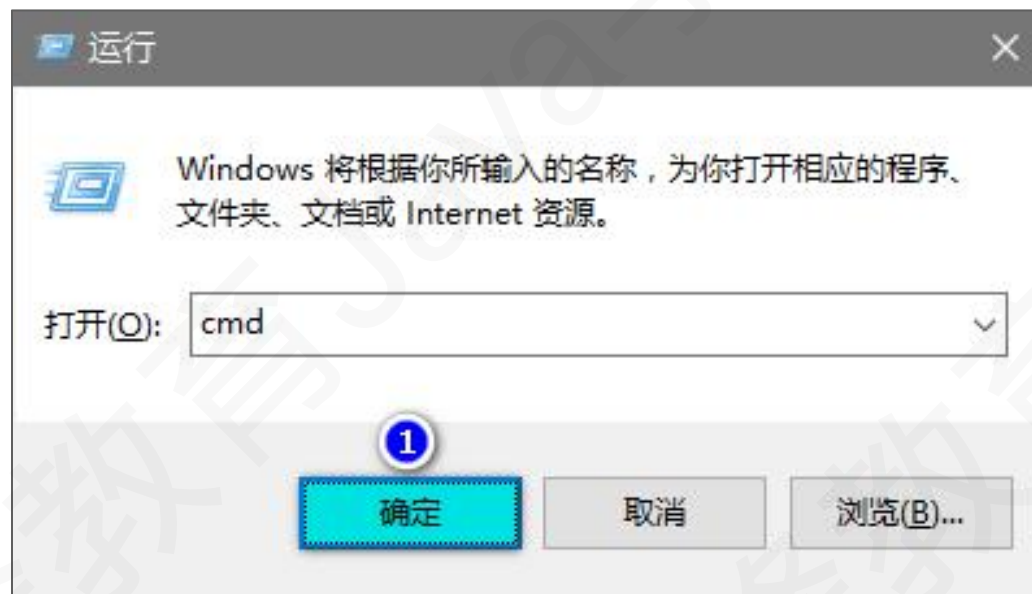
配置环境变量

“我的电脑”（“此电脑”） -> 右键点击“属性” -> “高级系统设置”



DOS命令操作

- Windows键 + R 快捷呼出运行窗口



- 再输入cmd并回车打开DOS命令窗口

- 更换盘符: d:
- 查看当前目录下的文件及文件夹: dir
- 进入文件夹: cd 文件夹的名字
- 返回上一级目录: cd ..
- 清空屏幕: cls
- 删除文件: del 文件名
- 删除文件夹: rd 文件夹名称
- 退出: exit

第一个Java应用程序

- 创建以 .java 结尾的源文件:

First.java

- 编写第一个程序:

HelloWorld:
类的名字 (自定义)

main: 主方法 (程序的入口)

class: 类
(代码的容器)

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

System.out.print(" ");控制台输出语句:
(" "之间的为控制台输出的具体文本内容)

;; 英文的分号
(代表一行代码的结束)

- 编译:

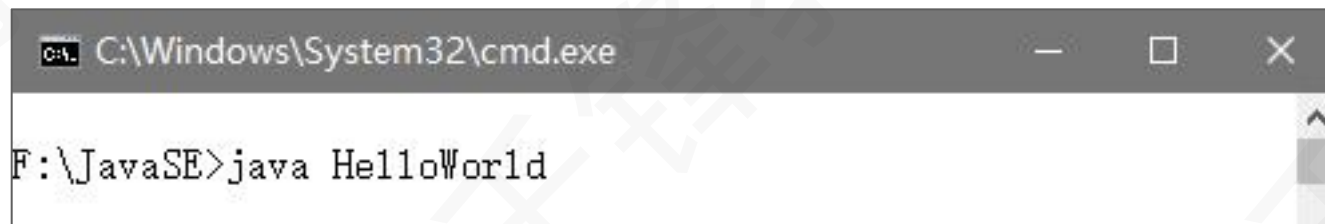
- javac 源文件名称 .java



```
C:\Windows\System32\cmd.exe  
F:\JavaSE>javac First.java
```

- 运行:

- java 类名



```
C:\Windows\System32\cmd.exe  
F:\JavaSE>java HelloWorld
```

- 同一个源文件中可以定义多个类。
- 编译后，每个类都会生成独立的 .class 文件。
- 一个类中，只能有一个主方法，每个类都可以有自己的主方法。
- public 修饰的类称为公开类，要求类名必须与文件名称完全相同，包括大小写。
- 一个源文件中，只能有一个公开类。

Package (包)

- 作用：类似于文件夹，用于管理字节码（.class）文件。
- 语法：package 包名;
- 位置：必须写在源文件的第一行。
- 带包编译：javac -d . 源文件名称.java （自动生成目录结构）
- 带包运行：java 包名.类名 （包名+类名又称全限定名）
- 采用域名倒置的规则：www.baidu.com.cn -> cn.com.baidu.xxx
- 例如：cn.com.company.department.group.project.module.XxxClass

编码规范 (1) 书写格式

- 层级之间必须缩进 (Tab: 一个制表位) 。
- 一行只写一句代码。

```
package demo;

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


编码规范 (2) 代码注释

- 单行注释:

```
// 单行注释
```

- 多行注释:

```
/* 多行注释 */
```

- 文档注释: (生成外部文档: javadoc -d . HelloWorld.java)

```
/** 文档注释 */
```

- 注: 注释不参与编译。

编码规范 (3) 标识符命名

- 语法规定：
 - 可以由：字母、数字、_、\$ 组成，但不能以数字开头。
 - 不能与关键字、保留字重名。
- 约定俗成：
 - 望文生义、见名知义。
 - 类名由一个或多个单词组成，每个单词首字母大写（pascal）。
 - 函数名、变量名由一个或多个单词组成，首单词首字母小写，拼接词首字母大写（camel）。
 - 包名全小写，只可以使用特殊字符 “.”，并且不以 “.” 开头或结尾。
 - 常量全大写，多个单词用 _ 连接。

- Java的由来：
 - 1995年推出的面向对象的编程语言。三个平台Java SE、Java EE、Java ME，2014年发布的JDK8是商用主流版本。
- Java语言的特点：
 - 面向对象、简单性、跨平台性。
- Java的运行机制：
 - 先编译、再解释。
- 第一个应用程序：
 - 定义.java结尾的源文件，书写源代码，javac 编译，java 运行。
- package 包：
 - 定义在源文件的首行，用于管理.class文件。
- Java的语言规范：
 - 书写格式、代码规范、三种注释、标识符命名。

THANK YOU



做真实的自己，用良心做教育