Java概述与环境搭建

Author：zhangzhang

Version：1.0

一、引言

1.1 什么是程序

> 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。

二、Java的历史与特点

2.1 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(Stadard 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

2.2 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）。

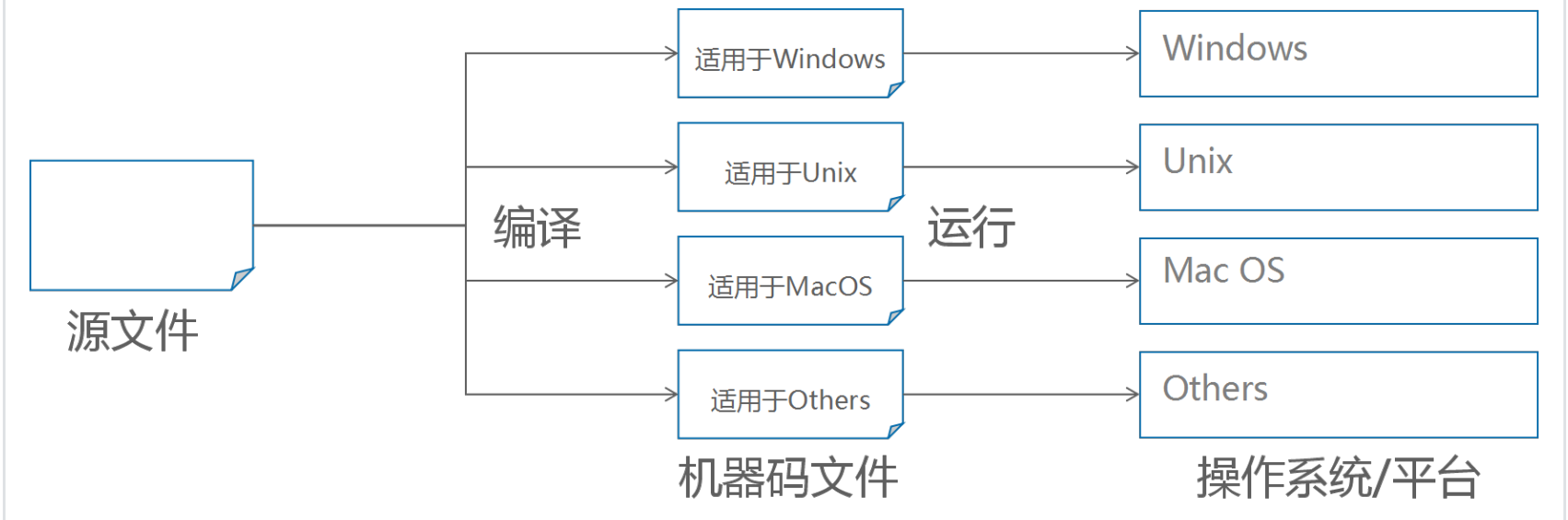
三、Java的运行机制【 重点 】

3.1 计算机的执行机制

3.1.1 编译执行

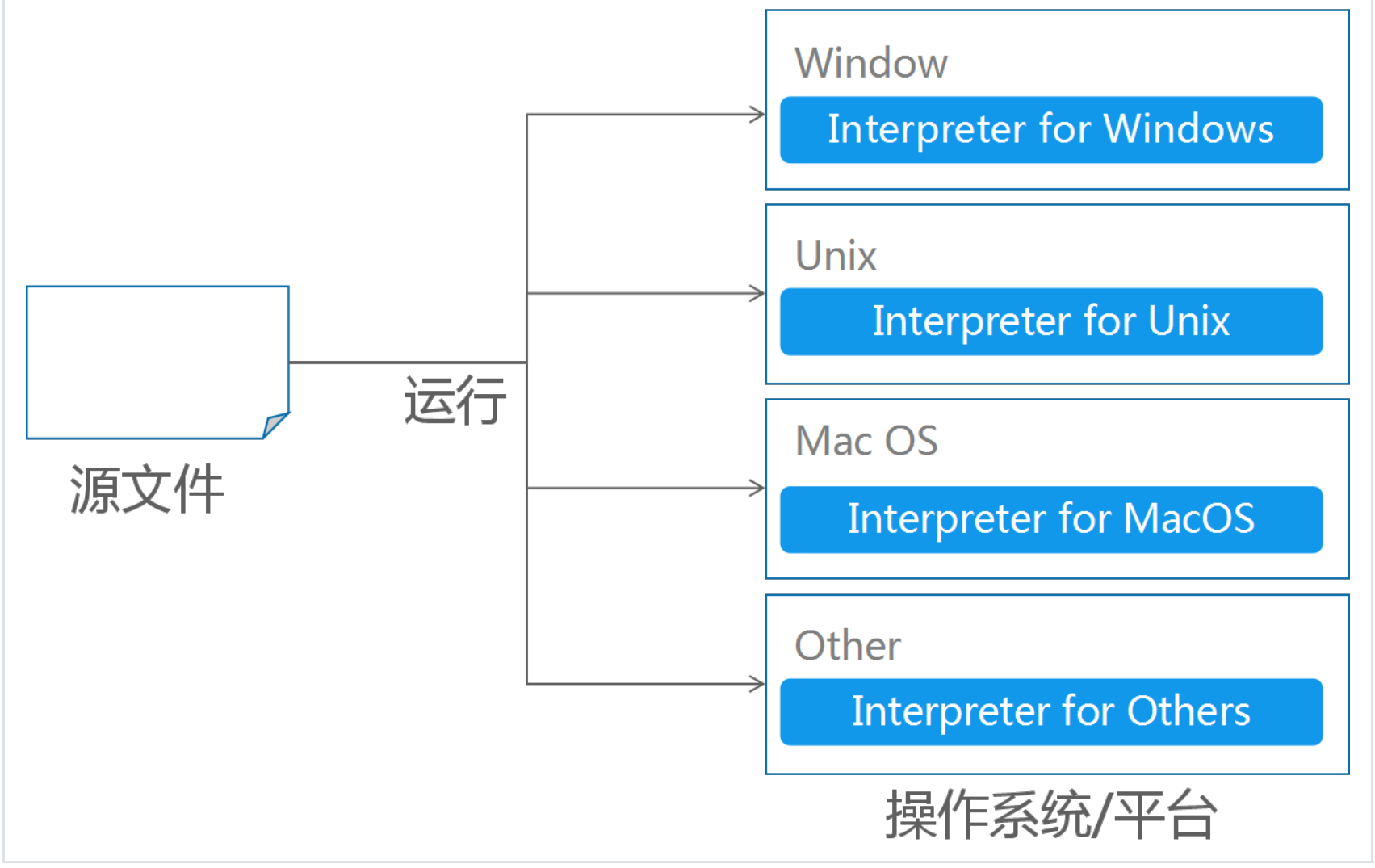
> - 将源文件编译成平台相关的机器码文件，一次编译，多次执行。

- 执行效率高，不可跨平台。



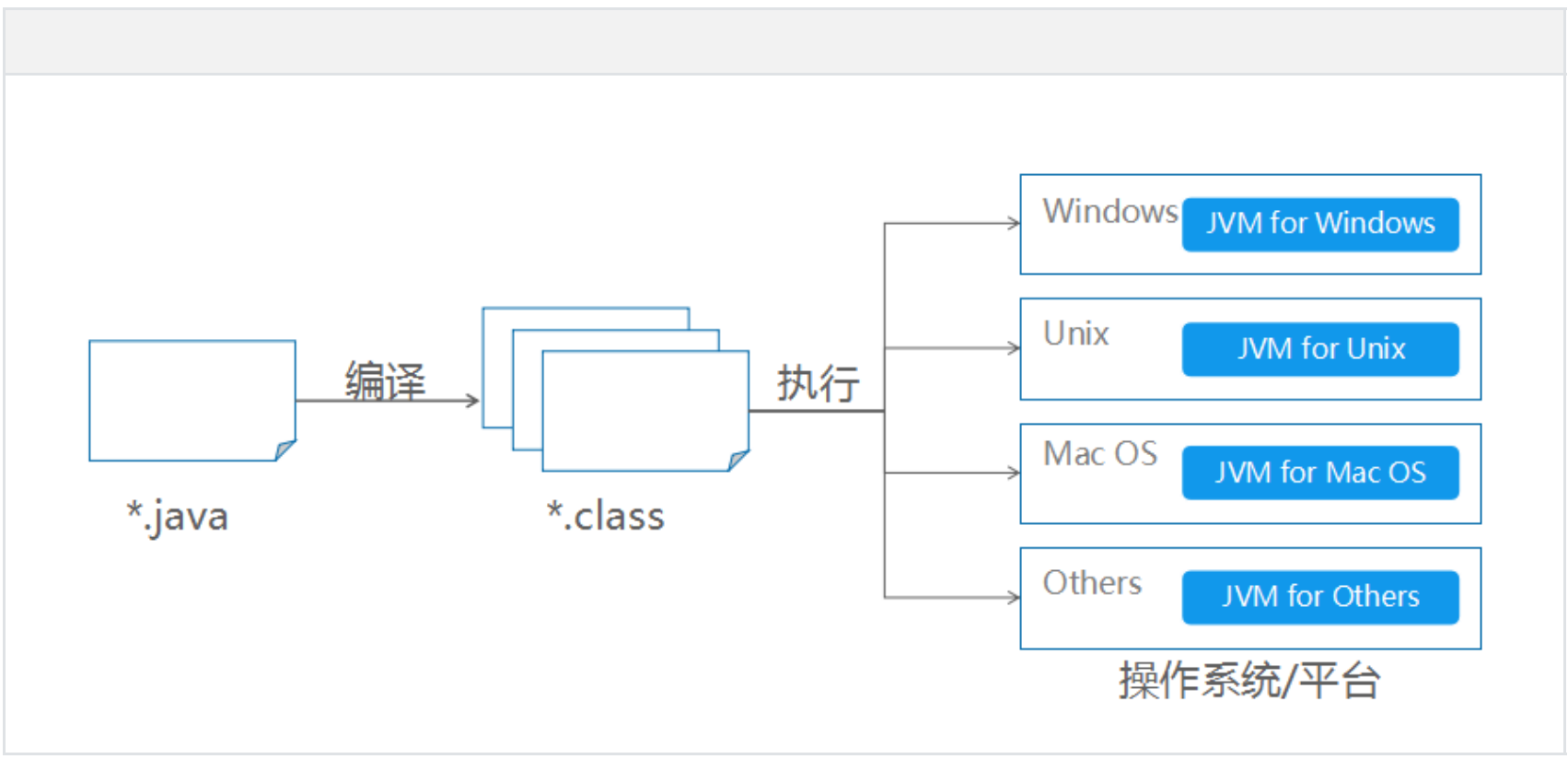
3.1.2 解释执行

- 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



3.2 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。



3.3 名词解释

- 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）。

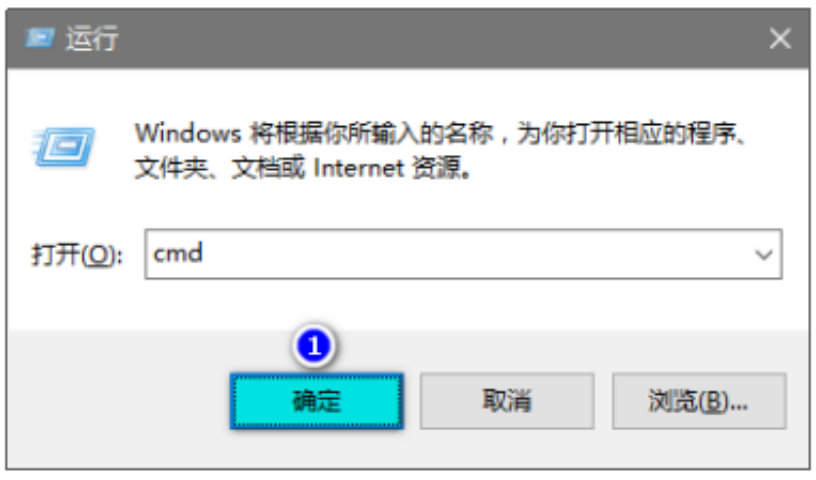
四、配置环境变量

4.1 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”。

4.2 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。



常用DOS命令：

- 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

五、第一个应用程序【 重点 】

5.1 第一个Java应用程序

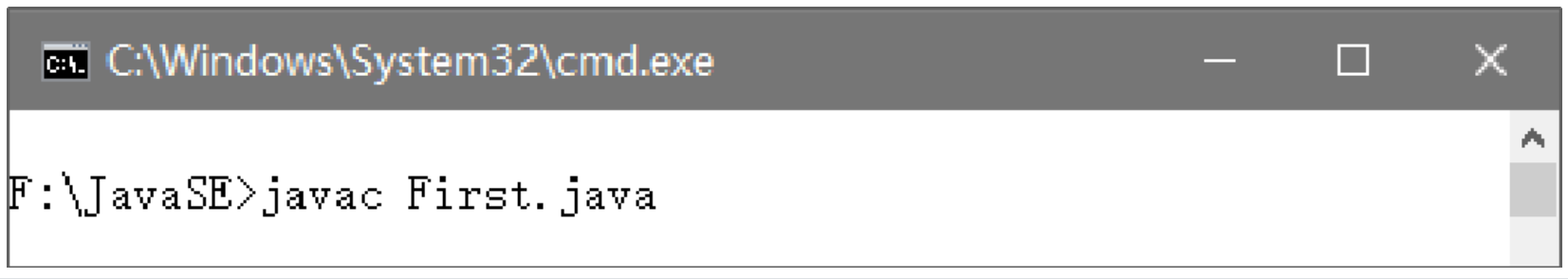
创建以 .java 结尾的源文件：

编写第一个程序：

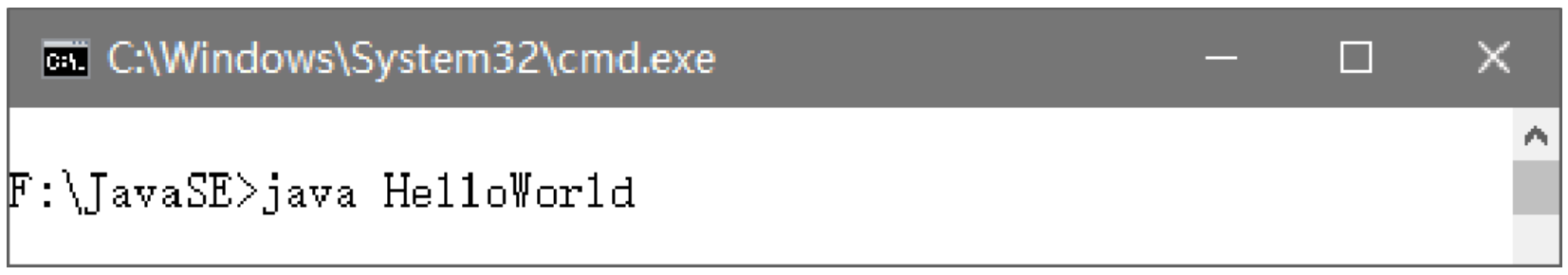

```
class HelloWorld{
  public static void main(String[] args){
    System.out.print("Hello Everyone");
  }
}
```

5.2 编译与运行

> 编译：javac 源文件名称 .java



> 运行：java 类名



5.3 类的阐述

> - 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。

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

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

5.4 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。

```
package com.qf.bj.class1.basic;

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

六、Java的语言规范

6.1 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");
    }
}
```

6.2 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。

```
package com.qf.chapter1.demo;

/**
文档注释
这个类旨在介绍Java中的注释方式有哪些
*/
public class TestAnnotation{//HelloWorld（帕斯卡） helloWorld（驼峰）  HELLO_WORLD      GetElementById
//getElementById     GET_ELEMENT_BY_ID

  /**
    这个方法代表程序的入口，即为书写逻辑代码的位置
  */
  public static void main(String[] args){
```

```
    //以下代码代表控制台输出一句文本
    System.out.print("HelloWorld");


     /*
     多行注释的开始
     .....
     .....
     .....
     多行注释的结束
     */
    System.out.print("Hello Everyone");

  }
}
```

6.3 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。