

第2章 Java入门引导

2.1 什么是程序

- **定义**：程序是一组有序的指令，告诉计算机如何执行特定的任务。
- **特点**
 - ：
 - 由代码构成，通常使用编程语言编写。
 - 需要编译或解释后才能运行。
 - 具有明确的目标和功能。

2.2 Java诞生小故事

- **背景**：1990年代初期，Sun Microsystems（现为Oracle）公司开发Java语言，最初目的是为了简化嵌入式设备的编程。
- **目标**：创建一种跨平台、安全且高效的编程语言。
- **发展**：Java迅速成为互联网开发的首选语言之一。

2.3 Java技术体系平台

- **核心平台**
 - ：
 - **JDK (Java Development Kit)**：包含开发工具和库，用于编写、编译和调试Java程序。
 - **JRE (Java Runtime Environment)**：运行Java程序所需的环境。
 - **JVM (Java Virtual Machine)**：解释并执行字节码的虚拟机。

2.4 Java重要特点

- **跨平台性**：一次编写，到处运行。
- **面向对象**：支持类、对象和继承等概念。
- **安全性高**：内置安全机制，防止恶意代码攻击。
- **自动内存管理**：垃圾回收器自动处理内存释放。

2.5 Java的开发工具

- **选择工具**
 - ：
 - **Eclipse**：开源、跨平台的集成开发环境（IDE）。
 - **IntelliJ IDEA**：功能强大的Java IDE，适合专业开发者。
 - **NetBeans**：另一个流行的开源Java IDE。

2.6 Java运行机制及运行过程

- **编译与解释**
 - ：
 - 源代码 → 编译器 → 字节码 → JVM → 执行。

- **JVM作用**：将字节码转换为机器指令，实现跨平台性。

2.7 什么是JDK, JRE

- **JDK**：包含JRE和开发工具（如编译器、调试器）。
- **JRE**：仅包含运行Java程序所需的环境，不包括开发工具。

2.8 下载、安装JDK

- 步骤
：
 - 访问Oracle官网或OpenJDK网站下载适合操作系统的JDK版本。
 - 双击安装包，按照提示完成安装。

2.9 配置环境变量 path

- **为什么配置path**：使系统能找到Java命令（如javac、java）。
- 步骤
：
 1. 找到JDK的安装路径，通常是 `C:\Program Files\Java\jdk-x.x.x_x`。
 2. 添加该路径及其bin目录到系统PATH环境变量。

2.10 安装JDK的练习

- 验证安装
：
 - 打开命令提示符，输入 `javac -version` 和 `java -version`，确认显示正确版本信息。

2.11 Java快速入门

- **第一个程序**：

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

- 运行步骤
：
 1. 编写代码并保存为 `HelloWorld.java`。
 2. 使用命令 `javac HelloWorld.java` 编译。
 3. 运行生成的字节码文件：`java HelloWorld`。

2.12 注释

- **单行注释**：使用 `//`，适用于简短解释。
- **多行注释**：使用 `/* */`，适合较长解释。
- **文档注释**：使用 `/** */`，用于生成API文档。

课前准备

jdk安装包

示例代码

教学方案设计

导入

1. 什么是编程？

组个词：编写程序，所以编程就是编写程序

2. 为什么要有编程

编程就是用来解决人们生活中的实际问题的，提高人们的日常生活质量

3. 那什么是程序呢？

程序就是一系列有序的命令的集合，比如美团，饿了么，滴滴，微信，网易云等，都是程序，那什么叫有序命令的集合呢？举个例子，美团的程序执行顺序就是商家在平台上传商品，用户在平台选择商家，选择好自己想要的商品，添加到购物车，填写地址后付款购买，骑手接收到订单，去商家取餐，送到用户手中，以上这就是一套完整的流程，这个顺序可以有变化吗？比如骑手可不可以先取餐？骑手都不知道去哪里取，取什么，送到哪里，怎么可以呢。所以程序必须要有一定的顺序才可以。

4. 那什么是编程语言呢？

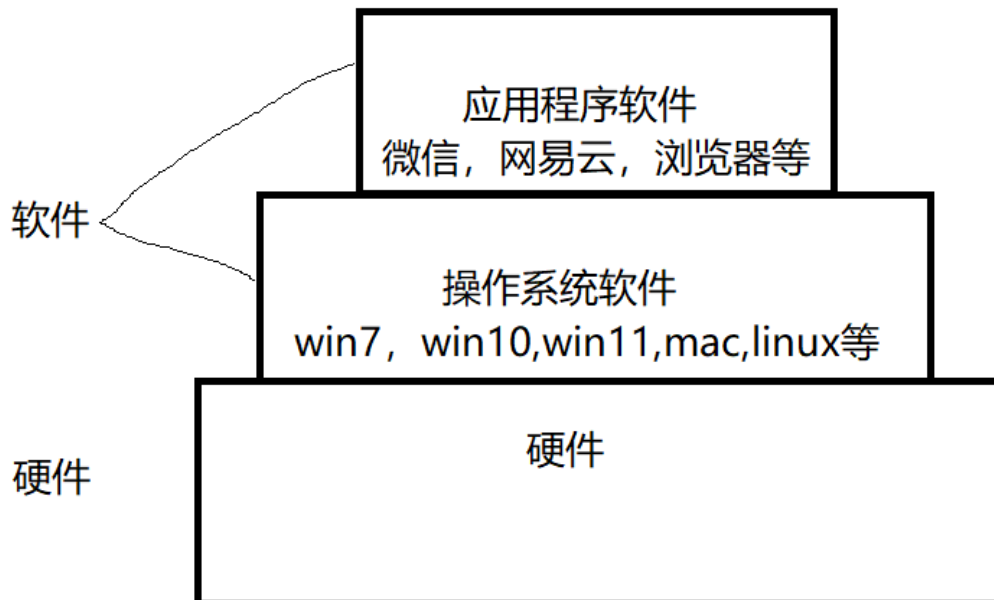
编程语言最重要的是什么？是编程吗？不是，是语言！

语言的本质是用来干嘛的？用来交流的，学会英文让我们可以和外国人相互交流，而外国人学会中文可以让他们和中国人进行交流，所以我们学会编程语言后可以让我们和计算机进行交流，我们的话计算机是听不懂的，所以我们要学习一个计算机能通懂得语言告诉计算机我们要让他去做什么。

那不同的编程语言我们怎么进行选择呢？

不同编程语言擅长的领域不同。比如，C 和 C++ 更贴近计算机底层，运行速度较快，因此常被用于项目中需要与计算机底层交互的部分。Java 具备“一次编译，到处运行”的跨平台特性，在开发多平台项目时优势比较明显，而多数服务器采用 Linux 系统，Java 通常作为服务器后端开发语言或管理系统的开发语言，可以更高效的适配服务器环境，也更加稳定。

5. 计算机的三层基本结构



6. 你知道哪些编程语言呢?

R语言, python, Java, C, C++等

7. 不同类型的编程语言的执行过程

- **编译型编程语言**: 在执行前需要通过编译器将源代码转换为机器码, 如 C、C++、Go 等, 优点是执行效率高, 缺点是开发调试周期较长。
- **解释型编程语言**: 要有一个编译器对源代码进行编译, 在执行时由解释器逐行解释并执行源代码, 如 Python、JavaScript、Ruby 等, 优点是开发调试方便, 可跨平台, 缺点是执行效率相对较低。

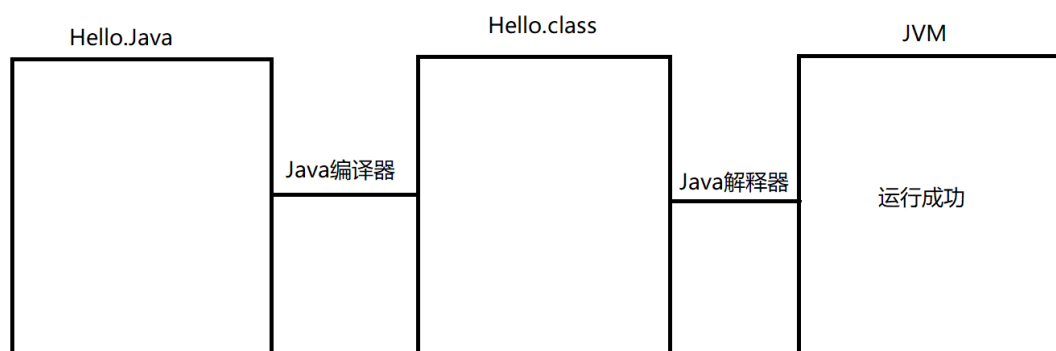
8. 有了编译类型的语言后为什么还要有解释类型呢?

编译类型的语言确实可以很方便的执行, 但是有一个很大的问题, 当操作系统不同时, 因为底层环境发生变化, 所以代码也需要进行相应的修改才能适应新环境, 这就是底层操作系统的不同造成代码不同

而解释类型的语言, 通常会在操作系统上安装不同版本的软件(虚拟机), 当我们将源代码编译成一个其他类型的文件(中间过渡的文件)后, 拿着这个编译后的文件在任何一个不同操作系统的软件(虚拟机)上都可以运行, 不需要再改变他的代码, 因为压根就用不到代码, 只需要那个中间的文件就可以完成运行, 这样就屏蔽了底层操作系统的差异, 就做到了跨平台运行。

9. Java的执行流程

因为Java属于解释型语言, 所以第一步要有一个编译器, 将源代码编译成那个中间文件(.class的字节码文件) 第二步使用Java解释器将.class的字节码文件放在不同操作系统的软件(java虚拟机:JVM)中执行



所以我们要先有一个编译器和一个解释器(用来将我们的源代码翻译成解释器能看懂的东西)才能运行Java的代码(Java解释器和JVM)，好消息是，我们不用一个一个下载，Java的编译器一般放在Java开发工具包（JDK）里面，已经包含了Java运行时环境和Java虚拟机，所以我们直接下载一个JDK（Java开发工具包）就可以了，里面什么都有。

10. 双击安装jdk，然后配置环境变量（演示）

学完本课后，可以独立完成以下任务

目标

1. 完成JDK的安装和配置
2. 编写一个完整的Java源文件
3. 在命令行完成Java运行的全过程
4. 输出语句
5. 输出不同类型的数据
 1. 数字 直接输出
 2. 算式 直接输出
 3. 字符 单引号
 4. 字符串 双引号
 5. 换行的两种方式

前测+参与式学习

1. 我的第一个Java程序

在D盘新建一个文件夹，名字叫做javapro，然后在文件夹中新建一个文本文档，文件名字起名为Hello.java

文件的名字分为两个部分，点之前的叫做文件名，点之后的叫做文件扩展名，扩展名表示的是文件的类型，比如 .txt 是文本文档，.exe 是windows的可执行文件，Java的源文件是以 .java 为扩展名的文件，文件名也是有命名规则的。

文件是由类组成的，类也是Java源文件的基本组成部分，首先新建一个公共类型的类，起个名字叫做Hello，那么我们的这个Java源文件的名字也必须叫Hello.java，否则就会报错。

```
public class Hello{  
  
}
```

2. 整个.java的源文件是从哪里开始执行的呢？

是从主方法开始的，大多数的编程语言都会有一个主函数的概念，但是在Java中一般叫做方法，主函数就是程序的入口，编译器会先从主函数开始，进行方法的调用，然后在从上到下顺序执行我们方法中的语句

3. 什么叫语句？

语句可以看成是一条命令，这个命令以分号结尾，比如输出语句（让程序在屏幕上显示要输出内容的命令）

4. 添加主方法和输出语句

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

5. 让程序运行起来

1. 第一步：打开命令提示符窗口

win+R, 打开运行框, 输入 `cmd` 回车打开命令提示符窗口, 打开后我们可以看到这个命令提示符窗口是由一个默认路径的, 一般是C开头, 也就是在C盘的某个目录中

2. 第二步：编译源文件

在命令提示符窗口中输入 `D:` 切换到D盘目录下

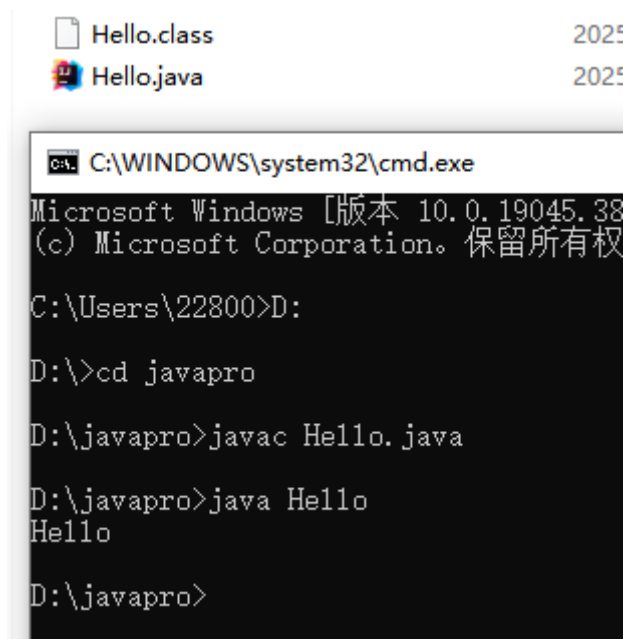
输入 `cd javapro` 进入到我们创建Java源文件的地方

使用 `javac Hello.java` 命令对源文件进行编译

这条命令完成后, 如果没有出现任何东西, 就说明编译成功了, 可以进行下一步, 我们可以打开D盘的javapro文件夹查看一下, 可以看到文件夹中多出了一个 `.class` 的字节码文件 (中间文件)

3. 第三步：运行 `.class` 的字节码文件

输入 `java Hello` 即可运行, 程序会输出一个Hello, 运行完毕



如上图所示即可

4. 将 `System.out.print("Hello");` 中的 `Hello` 删除掉, 然后在里面写一些自己喜欢的东西, 重复刚才的步骤, 观察输出的结果有什么不同

5. 数字类型：直接输出 (把双引号也删掉)

6. 算式类型：直接输出 (因为算式的结果也是数字)

7. 字符类型：什么叫字符?

组个词, 字母, 符号, 单个的字母或者符号都是一个字符

广义上来说, 只要键盘能打出来的东西都叫做字符, 空格能打出来, 所以单个的空格是字符, 回车也能打出来, 所以回车也是字符, 一会来说一下回车怎么打出来。

8. 字符类型的标志是单引号 `''`，电脑看到第一个单引号认为这是字符的开始，看到第二个认为是字符的结束，将内容括在单引号内就可以输出了（演示）

9. 尝试在单引号内输出两个或者两个以上的字符

结果不符合我们的预期，所以两个或者两个以上的字符就不能再使用单引号进行输出了

10. 一串字符连在一起，叫做字符串

字符串的标志是双引号 `""`，电脑看到第一个双引号认为这是字符串的开始，看到第二个认为是字符串的结束，将内容括在双引号内就可以输出了（演示）

11. 尝试输出两行，第一行输出abc第二行输出def

输出两句可不可以

```
System.out.print("abc");  
System.out.print("def");
```

很明显，不行

12. 第一种换行方式：转义字符 `\n` 转义字符属于字符的一种，可以放在单引号里也可以放在双引号里

```
System.out.print("abc\n");  
System.out.print("def");
```

第二种方式

```
System.out.println("abc");  
System.out.println("def");
```

后测

1. 后测题目

一、选择题（每题2分，共20分）

- 下列哪种编程语言的运行需要JVM（Java虚拟机）支持？
 - Python
 - Java
 - C
 - Go
- Java源代码文件的扩展名是？
 - .class
 - .exe
 - .java
 - .txt
- `System.out.print("Hello\nworld")` 的输出结果为？
 - HelloWorld
 - Helloworld
 - Hello World（中间有换行）
 - HelloWorld（中间无换行）

4. Java的主方法（程序入口）的固定写法是？
- A. `public static void main()`
 - B. `public static void Main(String[] args)`
 - C. `public static void main(String[] args)`
 - D. `public void main(String[] args)`
5. 下列哪个符号用于表示字符类型？
- A. `""`
 - B. `' '`
 - C. `<>`
 - D. `()`
6. 以下哪种方式可以实现换行输出？
- A. `System.out.print("\t")`
 - B. `System.out.println()`
 - C. `System.out.print("\n")`
 - D. `System.out.print(" ")`
7. Java的编译命令是？
- A. `java Hello.java`
 - B. `javac Hello.class`
 - C. `javac Hello.java`
 - D. `jvm Hello`
8. 若Java源文件中的类定义为 `public class MyClass`，文件名应该是？
- A. `MyClass.txt`
 - B. `myClass.java`
 - C. `MYCLASS.java`
 - D. `MyClass.java`
9. 以下哪种编程语言是编译型语言？
- A. Python
 - B. JavaScript
 - C. C++
 - D. Ruby
10. `System.out.print(3 + 4)` 的输出为？
- A. "7"
 - B. 7
 - C. 3 + 4
 - D. 报错

二、填空题（每空1分，共15分）

1. Java源文件经过编译后生成的文件扩展名是_____。
2. 主方法的参数是一个字符串数组，写作 `String[]` _____。
3. 输出语句 `System.out.print("A");` 的输出结果是_____（不换行/换行）。
4. 在Java中，单引号用于表示_____类型。
5. 输出 "Hello" 后换行的两种方式是：①_____ ② `System.out.println("Hello")`。
6. `System.out.print('A')` 的输出是_____。
7. 如果要运行一个名为 `Test.class` 的文件，命令是 _____ `Test`。
8. 字符串必须用_____符号包裹。
9. 数学表达式 `5 * 2` 的输出结果是_____。
10. 转义字符 `\n` 的功能是_____。

三、判断题（每题1分，共10分）

1. Java属于纯编译型语言。
 2. 主方法 `main` 可以省略不写。
 3. JDK中包含了JVM和编译器。
 4. `System.out.println()`; 会输出一个空行。
 5. 字符 `'AB'` 是合法的。
 6. 文件名必须与类名完全一致，包括大小写。
 7. `javac` 命令用于运行Java程序。
 8. `System.out.print("3 + 4")` 将输出 `7`。
 9. 环境变量的作用是为了让系统找到JDK工具。
 10. Java程序直接从类的第一行代码开始执行。
-

四、编程题（每题5分，共25分）

1. 编写一个Java程序，在控制台输出两行内容：
第一行：**I love Java**
第二行：**It's fun!**
2. 补全代码，输出字符 `A` 和字符串 `Hello`：

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

3. 写出以下代码的输出结果：

```
System.out.print("1");  
System.out.print("2\n3");  
System.out.println("4");
```

4. 设计程序输出如下内容（星号排成直角三角形）：

```
*  
**  
***  
****
```

5. 写出一个Java程序的完整结构（含类、主方法和输出语句）。
-

五、简答题（每题5分，共25分）

1. 简述解释型语言与编译型语言的区别。
2. 为什么需要配置环境变量？
3. Java的类名为什么必须与文件名一致？
4. 解释转义字符 `\n` 的作用，并举例说明。
5. 说明 `System.out.print()` 和 `System.out.println()` 的区别。

答案

一、选择题

1. B
2. C
3. C
4. C
5. B
6. B
7. C
8. D
9. C
10. B

二、填空题

1. `.class`
2. `args`
3. 不换行
4. 字符
5. `\n`
6. A
7. `java`
8. 双引号
9. 10
10. 换行

三、判断题

1. ×
2. ×
3. √
4. √
5. ×
6. √
7. ×
8. ×
9. √
10. ×

四、编程题

- 1.

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("I love Java");  
        System.out.print("It's fun!");  
    }  
}
```

2.

'A' 和 println

3.

```
12  
34
```

4.

```
System.out.println("*");  
System.out.println("**");  
System.out.println("***");  
System.out.println("****");
```

5.

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

五、简答题

1. **解释型语言**运行时逐行解释代码（如Python），**编译型语言**需先转换为机器码再执行（如C++）。
2. **环境变量**帮助系统找到DK工具（如 `javac` 和 `java` 命令）。
3. **Java规定**若类声明为 `public class`，文件名必须与类名完全一致。
4. `\n` 是换行符，例如 `System.out.print("A\nB")` 输出：

```
A  
B
```

5. `print()` 不换行，`println()` 输出后自动换行。