

1. 计算机基础

1.1. 计算机

电子计算机，俗称电脑，是一种能够按照程序运行，自动、高速处理海量数据的现代化智能电子设备。由硬件和软件所组成，没有安装任何软件的计算机称为裸机。

1.2. 硬件和软件

1.2.1. 硬件

1. **cpu** 是一台计算机的运算核心和控制核心。其功能主要是解释计算机指令以及处理计算机软件中的数据。CPU 由运算器、控制器、寄存器、高速缓存及实现它们之间联系的数据、控制及状态的总线构成。作为整个系统的核心，CPU 也是整个系统最高的执行单元，因此 CPU 已成为决定电脑性能的核心部件，很多用户都以它为标准来判断电脑的档次。CPU 的主频,即 CPU 内核工作的时钟频率 (CPU Clock Speed)。该时钟速度越快，时间段内执行的指令就越多，目前酷睿是 2G-3G (1G 是 1000Mhz) 例如，2MHz (200 万次/秒)，3GHz (三十亿次/秒)。
2. **内存**是数据与 CPU 进行沟通的桥梁。计算机中所有程序的运行都是在内存中进行的，内存 (Memory) 也被称为内存存储器，其作用是用于暂时存放 CPU 中的运算数据，以及与硬盘等外部存储器交换的数据。
3. **存储设备**：内存不能长久保存数据（非 ROM），断电时信息就丢失，程序和数据都会被保存在存储设备中（例如硬盘），当程序和数据需要使用时，再加载进内存，注意：内存读取和写入的速度要比存储设备快的多。
存储设备主要有：磁盘驱动器，光盘驱动器 (cd dvd) USB 闪存 (U 盘)。

1.2.2. 软件

软件的概述：按照一定顺序组织的计算机数据与指令的集合。

软件 = 数据+指令+文档

1. 系统软件
例如：Windows xp Linux Unix
2. 应用软件
例如：QQ、WPS、Photoshop、微软的 Office

软件开发：制作软件。

软件的本质：提高人与计算机交流效率，方便人与机器交互。

1.3. 人机交互的方式:

1. 图形化界面: 简单直观, 容易操作, 用户的体验好。
2. dos 命令交互方式: 需要有一个控制台, 输入特定的指令, 让计算机完成一些操作。较为麻烦, 需要记住一些命令, 而且只能完成一些比较单一的功能。

控制台打开的步骤: 开始-->所有程序--->附件-->命令行提示符
或者开始->输入 cmd 即可。

1.4. 常见的 dos 命令

盘符: 进入指定的盘符下。

dir : 列出当前目录下的文件以及文件夹

md : 创建目录

rd : 删除目录 注意: rd 不能删除非空的文件夹, 而且只能用于删除文件夹。

cd : 进入指定目录

cd .. : 退回到上一级目录

**cd ** : 退回到根目录

echo "hello java">a.txt 写入文本到指定文件

type a.txt 显示文件内容命令

del : 删除文件 注意: 不能删除文件夹, 只能删除文件。

exit : 推出 dos 命令行

cls : 通知计算机将当前的命令清空

*****: 是通配符。

tab 命令补全。

方向键上下是找回上次你写过的命令。

2. Java 简介

语言起源

Java 是 SUN (Stanford University Network 斯坦福大学网络公司) 1995 年推出的一门高级编程语言。

Java 名称的来源 Java 最初被命名为 Oak, 目标在于家用电器等小型系统的编程语言, 随着市场的变化, SUN 着眼于网络应用的开发。Java 的名字的来源: Java 是印度尼西亚爪哇岛的英文名称, 因盛产咖啡而闻名。 机顶盒 高斯林 (高司令) 用户想看什么片就看什么...优酷

1995 年 5 月 23 日, JAVA 语言诞生 雏形

1996 年, 1997 分别推出 JDK1.0, JDK1.1 版

1999 年, JAVA 被分成 J2SE, J2EE, J2ME

2000 年, JDK1.4 发布

2004 年 9 月 30 日 18:00PM, J2SE1.5 发布, 成为 Java 语言发展史上的又一里程碑。

为了表示该版本的重要性, J2SE1.5 更名为 Java SE 5.0

2005 年, JAVA 版本正式更名为 JAVAE, JAVASE, JAVAME

2006 年 12 月, SUN 公司发布 JRE6.0

2009 年 04 月 20 日, Oracle 以 74 亿美元收购 Sun。取得 java 的版权。

2010 年 9 月, JDK7.0 已经发布, 增加了简单闭包功能。

2011 年 7 月, 甲骨文公司发布 java7 的正式版。

2013 年 8 月计算机语言使用排行榜

Position Aug 2013	Position Aug 2012	Delta in Position	Programming Language	Ratings Aug 2013	Delta Aug 2012	Status
1	2	↑	Java	15.978%	-0.37%	A
2	1	↓	C	15.974%	-2.96%	A
3	4	↑	C++	9.371%	+0.04%	A
4	3	↓	Objective-C	8.082%	-1.46%	A
5	6	↑	PHP	6.694%	+1.17%	A
6	5	↓	C#	6.117%	-0.47%	A
7	7	=	(Visual) Basic	3.873%	-1.46%	A
8	8	=	Python	3.603%	-0.27%	A
9	11	↑↑	JavaScript	2.093%	+0.73%	A
10	10	=	Ruby	2.067%	+0.38%	A

疑问: 学习 java 需要考证吗? (不需要考的, 英语四级证书)

2.2 java 语言重要特性之跨平台

2.2.1 跨平台

什么是跨平台(计算机系统), 跨平台概念是软件开发中一个重要的概念, 即不依赖于操作系统, 也不信赖硬件环境。一个操作系统下开发的应用, 放到另一个操作系统下依然可以运行。

2.2.2. Java 的跨平台

“write once run anywhere”

Java 的跨平台是相对于其他编程语言而言的, 通过 Java 语言编写的应用程序在不同的系统平台上都可以运行。一般的高级语言如果要在不同的平台上运行, 至少需要编译成不同的目标代码。在 windows 编译的 c 语言程序, 在 linux 下是不能运行的。当然在 linux

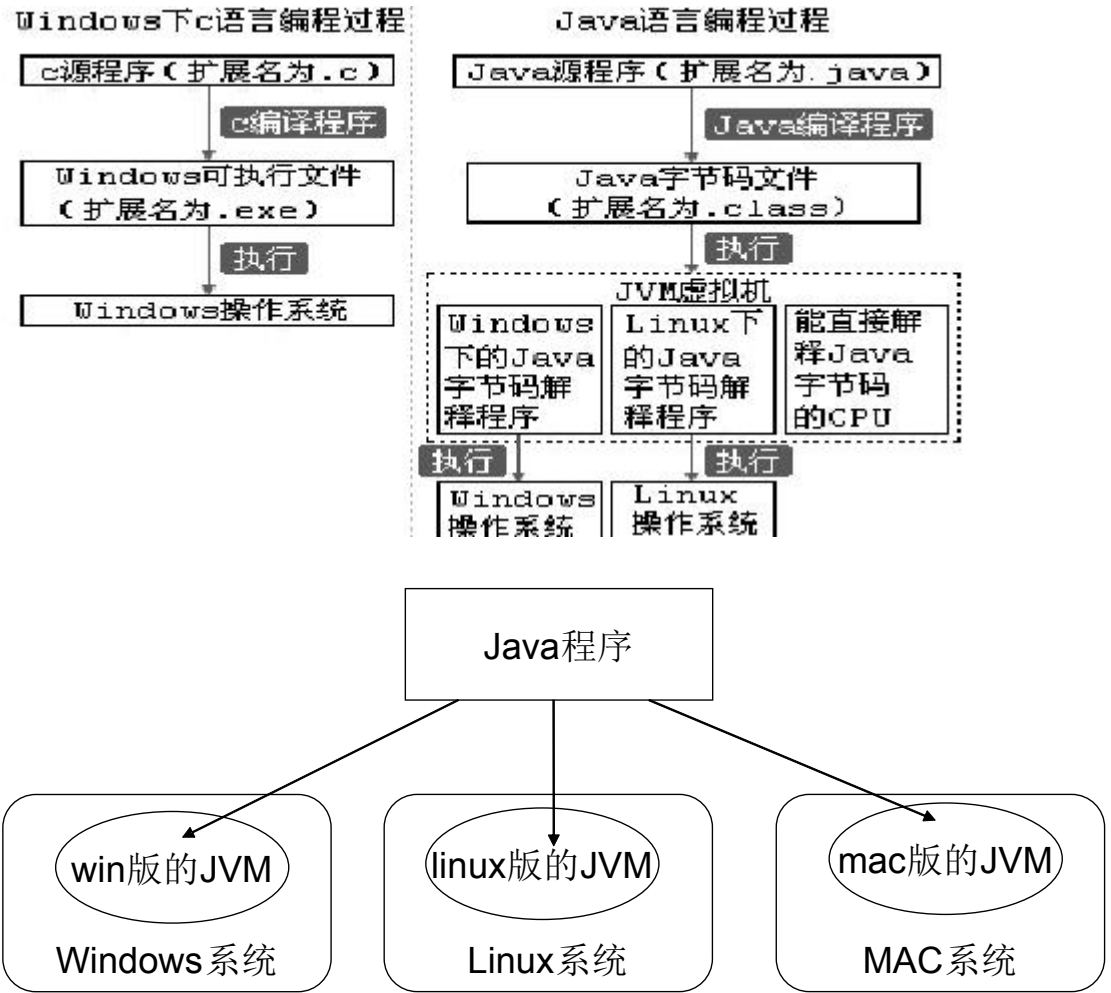
下编译的 c 语言程序在 windows 下也是不能运行的。而 Java 语言在不同平台上运行时不需要重新编译。

Java 跨平台的原理：Java 的跨平台是通过 Java 虚拟机（JVM）来实现的。

2.2.3. Java 虚拟机（JVM）

Java Virtual Machine 一个软件模拟的计算机。Java 源程序想象成我们的 C 源程序，Java 源程序编译后生成的字节码（.class）就相当于 C 源程序编译后的二进制可执行文件。JVM 虚拟机相当于计算机系统（操作系统+硬件环境），Java 解释器相当于 CPU。

.exe 文件直接在 windows 操作系统下运行，在 CPU 上运行的是机器码（.exe 文件）。.class 字节码文件是运行在 JVM 虚拟机下的，在 Java 解释器上运行的是 Java 字节码。Java 解释器相当于运行 Java 字节码的“CPU”，但该“CPU”不是通过硬件实现的，而是用软件实现的。Java 解释器实际上就是特定的平台下的一个应用程序。只要实现了特定平台下的解释器程序，Java 字节码就能通过解释器程序在该平台下运行，这是 Java 跨平台的根本。当前，并不是在所有的平台下都有相应 Java 解释器程序，这也是 Java 并不能在所有的平台下都能运行的原因，它只能在已实现了 Java 解释器程序的平台下运行。



2.3Java 的三大平台

JavaSE

Java SE (Java Platform, Standard Edition)。Java SE 以前称为 J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 包含了支持 Java Web 服务开发的类, 并为 Java Platform, Enterprise Edition (Java EE) 提供基础。

例子: 类似 QQ、扫雷、飞秋、凌波这样子的桌面应用。

Java EE

Java EE (Java Platform, Enterprise Edition)。这个版本以前称为 J2EE。企业版本帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的, 它提供 Web 服务、组件模型、管理和通信 API, 可以用来实现企业级的面向服务体系结构 (service-oriented architecture, SOA) 和 Web 2.0 应用程序。

Java ME

Java ME (Java Platform, Micro Edition)。这个版本以前称为 J2ME, 也叫 K-JAVA。Java ME 为在移动设备和嵌入式设备 (比如手机、PDA、电视机顶盒和打印机) 上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对动态下载的连网和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次, 就可以用于许多设备, 而且可以利用每个设备的本机功能。

3.环境的搭建

3.1JDK 简介:

JDK (Java Development Kit) 是 Sun Microsystems 针对 Java 开发员的产品。自从 Java 推出以来, JDK 已经成为使用最广泛的 Java SDK。JDK 是整个 Java 的核心, 包括了 Java 运行环境、Java 工具和 Java 基础类库。JDK 是学好 Java 的第一步。从 SUN 的 JDK5.0开始, 提供了泛型等非常实用的功能, 其版本也不断更新, 运行效率得到了非常大的提高。

3.2JDK 的安装

网址: <http://www.oracle.com/index.html> 、 <http://www.java.sun.com>

ORACLE® Sign In/Register for Account Help Select Country/Region Communities I am a... I want to... Search

PRODUCTS AND SERVICES SOLUTIONS DOWNLOADS STORE SUPPORT TRAINING PARTNERS ABOUT

Oracle Technology Network > Java

Java Developer
Database Administrator / Developer
System Admin / Developer
Architect

C-Level Executive
Chief Financial Officer
Chief Human Resources Officer
Chief Information Officer

Other Roles
Analyst
Investor
Job Seeker
Partner
Student
Midsize Company

Java SE 7 Update 10
Released 12/11/12

Java SE 6 Update 38
Released 12/11/12

Java Embedded 7 Update 10
Released 12/11/12

Java Embedded 6 Update 38
Released 12/11/12

End of Public Updates for Java SE 6

JDeveloper 11g and ADF Enterprise Pack for Eclipse

ORACLE® Sign In/Register for Account Help Select Country/Region Communities I am a... I want to... Search

PRODUCTS AND SERVICES SOLUTIONS DOWNLOADS STORE SUPPORT TRAINING PARTNERS ABOUT

Oracle Technology Network > Java

Java SE 7 Update 10
Released 12/11/12

Java SE 6 Update 38
Released 12/11/12

Java Embedded 7 Update 10
Released 12/11/12

Java Embedded 6 Update 38
Released 12/11/12

JavaFX 2.2.4
Released 12/11/12

Oracle Java Embedded Suite 7.0
Released 9/25/12

How to Get Started (FAST!) with JavaFX 2 and Scene Builder

Oracle Engineer Mark Heckler shows you how to hit the ground running with JavaFX and Scene Builder.

Posted 11/17/12 // Tags: java, JavaFX, RIA // Headlines Archive

Software Downloads View All Downloads

Top Downloads

Java SE

Java EE and GlassFish

JavaFX

Java ME

JDeveloper 11g and ADF

Enterprise Pack for Eclipse

NetBeans IDE

Pre-Built VM for Java Developers

here are the Java SE downloads in detail.

Java Platform, Standard Edition		
<p>Java SE 7u10</p> <p>This releases brings in key security features and bug fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release. JavaFX 2.2.4 is now bundled with the JDK on Windows, Mac and Linux x86/x64.</p> <p>Learn more</p> <p>"What Java Do I Need?" You must have a copy of the JRE (Java Runtime Environment) on your system to run Java applications and applets. To develop Java applications and applets, you need the JDK (Java Development Kit), which includes the JRE.</p>	<p>JDK</p> <p>DOWNLOAD</p> <p>JDK 7 Docs</p> <ul style="list-style-type: none"> Installation Instructions ReadMe ReleaseNotes Oracle License Java SE Products Third Party Licenses Certified System Configurations 	<p>JRE</p> <p>DOWNLOAD</p> <p>JRE 7 Docs</p> <ul style="list-style-type: none"> Installation Instructions ReadMe ReleaseNotes Oracle License Java SE Products Third Party Licenses Certified System Configurations

3.3JDK 目录介绍

1. bin 目录: 存放 Java 的编译器、解释器等工具(可执行文件)。
2. db 目录: JDK7 附带的一个轻量级的数据库, 名字叫做 Derby。
3. include 目录: 存放的调用系统资源的接口文件。
4. jre 目录: 存放 Java 运行环境文件。
5. lib 目录: 存放 Java 的类库文件。
6. src.zip 文件: JDK 提供的类的源代码。

安装 JDK 需要注意的事项:

1. 安装路径中不要包含中文。
2. 安装路径中不要包含空格。

3.4JRE 简

JRE Java Runtime Environment Java 运行环境。包括 Java 虚拟机 (JVM Java Virtual Machine) 和 Java 程序所需的核心类库等, 如果想要运行一个开发好的 Java 程序, 计算机中只需要安装 JRE 即可。

3.5 配置 path 环境变量

问题: 目前在控制台窗口下只能在 jdk 安装目录的 bin 目录下才能使用 jdk 的开发工具在控, 如果想控制台窗口下想要在任意目录下都可以使用 javac 与 java 或者 jdk 其他的开发工具那应该怎么做呢?

1. 在 dos 窗口敲上 javac



```
C:\Documents and Settings\Administrator>javac
'javac' 不是内部或外部命令, 也不是可运行的程序
或批处理文件。
```

我们在随意的一个目录敲 javac 或者 java 命令时候, 发现 dos 窗口显示这个不是内部命令, 然后我们在上面敲上 notepad 命令时, 打开了一个记事本的程序, 但是在当前目录下并没有存在 notepad.exe 的可执行文件, 那么为什么能启动呢?

错误原因: 原来在 dos 窗口上写上一个命令时候, 如果在当前目录下找不到指定的可执行文件, 那么它会根据 path 环境变量保存的路径去搜索是否存在这个可执行文件, 如果存在则启动该程序, 否则就没有这个命令。

2. 解决方法:

- 1、进入 jdk 的安装目录的 bin 目录。
- 2、在 dos 命令上敲 javac。(如果出现了以上的图信息则配置成功)

```

C:\Documents and Settings\Administrator>java
Usage: java [-options] class [args...]
           (to execute a class)
    or  java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
    -client      to select the "client" VM
    -server      to select the "server" VM
    -hotspot     is a synonym for the "client" VM [deprecated]
                 The default VM is client.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                 A ; separated list of directories, JAR archives,
                 and ZIP archives to search for class files.

```

3.为什么要配置 path 环境变量

因为在 jdk 下 bin 文件夹中有很多我们在开发中要使用的工具，如 java.exe, javac.exe, jar.exe 等,那么我们在使用时，想要在电脑的任意位置下使用这些 java 开发工具，那么我们就需有把这些工具所在的路径配置到系统的环境变量中，当我们使用时，系统可以帮我们找到这些命令。

4.如何配置 path 环境变量

我们将 jdk 下 bin 的路径进行复制

在我们电脑----- 属性----- 高级----- 环境变量----- path 这项
将 bin 的路径放置到 path 的最前边，别忘记加分号。

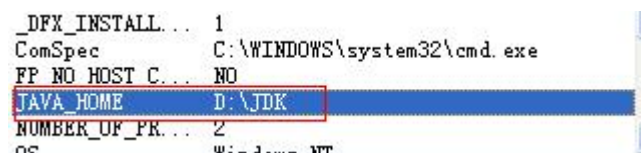




5.JAVA HOME 环境的配置

有时候可能需要更换 Jdk 的目录，但是经常修改 path 的值可能会不小心修改其他的路径，解决方法：

- 1、 创建一个 JAVA_HOME 的变量。



- 2、JAVA_HOME 的值为 JDK 的安装目录。
- 3、配置 path 路径: %JAVA_HOME%/bin。



以后需要更换 JDK 的目录时，只需要更换 JAVA HOME 的目录即可。

设置临时变量:

set : 查看所有的环境变量信息。

set 变量名：查看具体一个环境变量的值。

set 变量名=: 清空一个环境变量的值。

set 变量名=具体值 : 给指定变量定义具体值。

想要在原有环境变量值基础上添加新值呢？

首先，通过%变量名%操作符获取到原有环境变量的值。

然后加上新值后在定义给该变量名即可

举例：给 path 环境变量加入新值

set path=新值;%path%

注意：这种配置方式只在当前 dos 窗口有效。窗口关闭，配置消失。

4 . JAVA 程序入门

4. 1Hello Word 程序入门

1. 打开一个记事本在记事本里面写上 `class 类名 {}`
2. 一个类缺少主函数不能独立运行。
3. 可以独立运行了，但是没有效果。输出语句。

HelloWord 例子剖析：

`class java` 关键字，用于什么一个类、

`{}`：花括号表示范围。

`main` : 是主函数的如果（固定写法）。

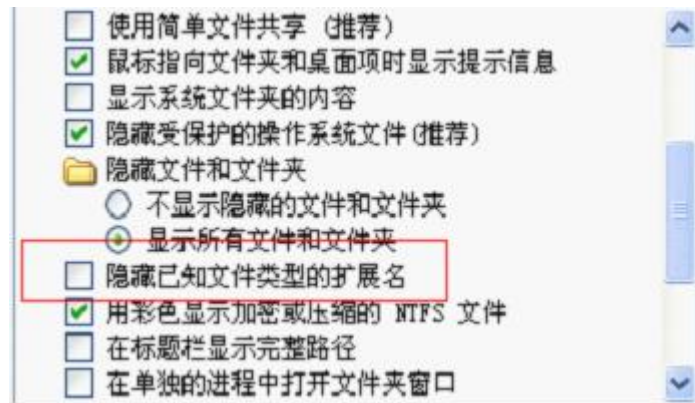
4.2 helloworld 的小细节

1. 缺少 `main` 方法

出现提示：缺少一个名称为 `main` 的方法。

```
C:\>java Demo
Exception in thread "main" java.lang.NoSuchMethodError: main
```

2. 源文件中的大小写要注意。初学者请牢记，第一个 `hello` 例子中只有 `String` 和 `System` 中两个字符是大写的，其他的都小写。
3. 配置 `path` 和 `classpath` 时，里面的路径是所有查找的目标的父目录，而不是目标自己。
4. `A.java.txt`
文本文件默认是 `.txt` 结尾的，注意要把隐藏已知文件类型扩展名去掉。



5. 大小写

只有类名 Demo, String 和 System 中三个个首字母是大写的。

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

6. path

看 path 环境变量配置，里面的路径是所有查找的目标的父目录，而不是目标自己。

```
C:\>javac Demo.java  
javac: 找不到文件: Demo.java  
用法: javac <options> <source files>  
-help 用于列出可能的选项
```

7. 写错类名，源文件名不存在或者写错，或者当前路径错误。

类文件名写错，或者类文件不在当前路径下，或者不在 classpath 指定路径下。

```
C:\>java Demo  
Exception in thread "main" java.lang.NoClassDefFoundError: Demo  
Caused by: java.lang.ClassNotFoundException: Demo  
    at java.net.URLClassLoader$1.run(URLClassLoader.java:202)  
    at java.security.AccessController.doPrivileged(Native Method)  
    at java.net.URLClassLoader.findClass(URLClassLoader.java:190)  
    at java.lang.ClassLoader.loadClass(ClassLoader.java:307)  
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)  
    at java.lang.ClassLoader.loadClass(ClassLoader.java:248)  
Could not find the main class: Demo. Program will exit.
```

8. javac 生成的.class 文件的名称取决于类名跟.java 名没有关系。

4.3 classpath 环境变量

1. classpath 的作用：作用是指定类搜索路径，要使用已经编写好的类，前提当然是能够找到它们了，一旦配置了 classpath 路径信息的时候，jvm 与 java 编译器都会根据 classpath 指定的路径去寻找 class 文件。
2. 如何设置当前目录 (.)
set classpath=.; 设置路径 与 set classpath=设置路径;. 有什么区别?
“.”在前面的时候，java 虚拟机会先从当前路径去寻找 class 文件，然后再到指定的路径去查找。如果”.”在后面，java 虚拟机则会先寻找指定好的路径,然后再在当前的路径去寻找。
设置 classpath 的时候要加上 ; 就会在当前目录下查找。

5.作业

1. 简述对 JVM, JRE, JDK 的理解。
JVM 虚拟机相当于计算机系统（操作系统+硬件环境）
jre = java 虚拟机 + 核心类库 (辅助 java 虚拟机运行的文件)
jdk = jre + java 开发工具
2. 简述环境变量 path 和 classpath 的作用?
搭建环境变量 path 可以在任意目录下运行对应目录的所有软件
3. 独立编写 hello world 程序，编译并运行。

```
class Demo{
    public static void main(String[]args){
        System.out.println("hello world");
    }
}
```
4. java 语言是否区分大小写? class 的作用,javac,java 的作用?
分、class 存放代码 javac 编译代码 Java 执行代码
5. 简述 main 方法(是什么，什么特点，怎么用，何时用?)
Main 主方法，一个类有且仅有一个主方法，一个类必须要用一个主方法，程序的入口