day01_前言、入门程序、常量、变量

一.开发前言

1.计算机基础知识

(1).计算机中数据的存储方式

计算机中的所有数据都是以二进制数的形式存储在内存中的，二进制数字上的每一位(0或1)，称为一个比特位(bit)。虽然数据都是以二进制数的形式存储在内存中的，但是字节才是计算机的最小存储单元，其中8个bit位(二进制位：0000 0000)称为1个字节，写成1byte或者1B。即计算机存储任何数据，都是以一个字节为基本单位，使用一个8位二进制数字(0000 0000)作为数据的最小存储单元。数据的大小除了字节，还有如下单位可以衡量：

8 bit	=	1 B
1024 B	=	1 KB
1024 KB	=	1 MB
1024 MB	=	1 GB
1024 GB	=	1 TB
1024 TB	=	1 PB
……		

(2).常用 DOS 命令

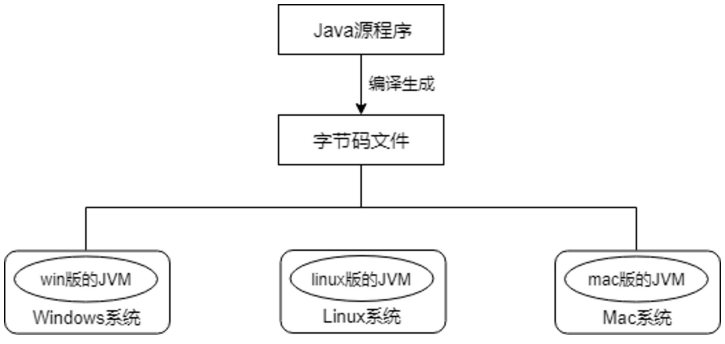
MS-DOS(Microsoft Disk Operating System)，称为磁盘操作系统。DOS系统的功能已经被集成到了Windows系统里面，叫做命令提示符。

命令提示符的启动：	Win+R，输入cmd回车(默认在用户主目录下启动)
切换盘符：	盘符名称：(D：切换到D盘)
进入文件夹：	cd 文件夹名称
进入多级文件夹：	cd 文件夹1\文件夹2\文件夹3
返回上一级：	cd ..
返回所在磁盘的根路径：	cd \
查看当前路径下的内容：	dir(directory的缩写)
命令自动补全：	tab
清屏：	cls(clear screen的缩写)
退出：	exit

2.Java语言开发环境搭建

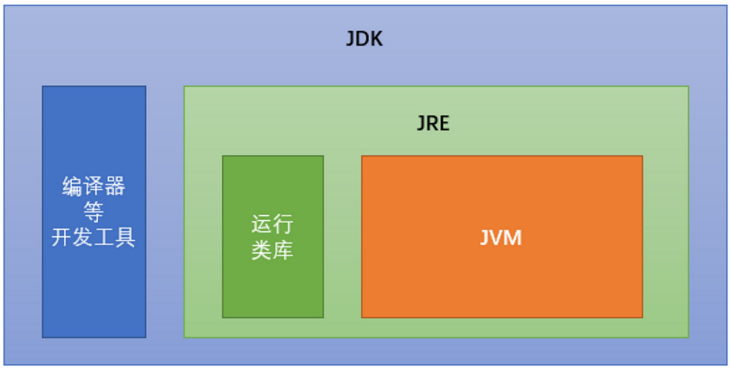
(1).Java语言的跨平台性

JVM(Java Virtual Machine)：Java虚拟机，是一种抽象化的计算机，通过在实际的计算机上仿真模拟各种计算机功能来实现的。Java虚拟机有自己完善的硬体架构，如处理器、堆栈、寄存器等，还具有相应的指令系统。如图所示，Java源程序(.java文件)通过编译生成字节码文件(.class文件)，字节码文件运行在JVM上，而JVM运行在操作系统上。虽然Java虚拟机本身不具备跨平台功能，但每个操作系统下都有不同版本的虚拟机。字节码文件可以不加修改地运行在各个版本的JVM上，以此来实现Java语言的跨平台性。



(2).JRE和JDK

- JRE(Java Runtime Environment)：是Java程序运行时的环境，包含JVM和运行时所需要的核心类库 。要运行一个已有的Java程序，那么只需安装JRE 即可。
- JDK(Java Development Kit)：是Java程序开发工具包，包含JRE和开发工具。要开发一个Java程序，那么必须安装JDK。



(3).JDK的下载与安装：见视频

(4).JAVA_HOME环境变量的配置

①.配置环境变量的作用

开发Java程序，需要使用JDK中提供的工具（如：编译器javac.exe、解释器java.exe等工具），这些工具在JDK的bin目录下，而且必须在DOS命令行下才能使用。要想在DOS命令行下使用这些工具，就要先进入到JDK的bin目录下进行使用，这个过程就会非常的麻烦。为了开发方便，我们想在DOS命令行的任意目录下都可以使用JDK的开发工具，则必须要配置环境变量，配置环境变量的意义在于告诉操作系统，我们使用的JDK开发工具在哪个目录下。

②.配置环境变量的步骤

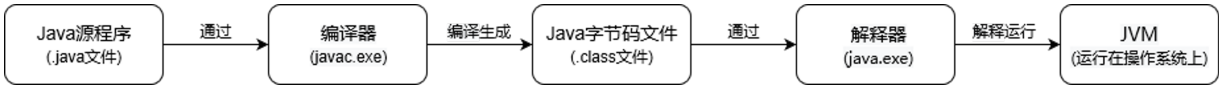
第一步：新建名为JAVA_HOME的系统变量，变量值输入JDK的安装目录：D:\Program Files\Java\jdk-9.0.4

第二步：选中Path系统变量，键入%JAVA_HOME%\bin，必须是英文格式（其中%JAVA_HOME%就代表上述JAVA_HOME的路径）

第三步：重启命令行，输入java、javac命令测试环境变量是否配置成功

二.HelloWorld入门程序

1.Java程序的开发步骤



(1).编写Java源程序

使用记事本、Notepad++，键入入门程序代码并将该文件保存为：HelloWorld.java　　注意：文件名必须和类名保持一致

(2).编译Java源文件

编译就是将我们编写的Java源文件通过javac编译器翻译成JVM认识的.class字节码文件。编译成功后，在Java源文件的保存目录下会生成字节码文件HelloWorld.class。（在DOS命令行中，进入Java源文件的目录，使用javac命令进行编译：javac HelloWorld.java ）

(3).运行Java程序

运行就是将编译生成的.class字节码文件交给JVM去运行，此时JVM就会去执行我们编写的程序了。（在DOS命令行中，进入Java源文件的目录，使用java命令进行运行：java HelloWorld ，运行时只使用类名就可以了，不能带后缀 .class）

2.HelloWorld入门程序说明　　（* 类是Java当中所有源代码的基本组织单位）

(1).注释：

- 单行注释：以'//'开始，以换行结束
- 多行注释：以'/*'开始，以'*/'结束

(2).关键字(keywords)：在程序中，Java已经定义好的、有特殊含义的、被保留的、不能随意使用的字符。

- 完全小写的字母
- 在增强版的记事本(例如Notepad++)或者一些IDE中，有特殊颜色

(3).标识符(identifier)：在程序中，我们自己定义的内容。比如类的名字、方法的名字和变量的名字等等，都是标识符。

①.命名规则：硬性要求

- 标识符可以使用26个英文字母(区分大小写)、0-9数字、$(美元符号)和 _(下划线)
- 标识符不能以数字开头
- 不可使用关键字作为标识符

②.命名规范：软性建议

- 类名规范：首字母大写，后面每个单词首字母大写（大驼峰式）
- 方法名规范：首字母小写，后面每个单词首字母大写（小驼峰式）
- 变量名规范：同方法名规范

三.常量

常量(const)：是指在Java程序中固定不变的数据。(即程序中一些具体、固定的数据值)

- 整数常量：所有的整数。例如：100、200、0
- 浮点数常量：所有的小数。例如：2.5、-3.14、0.0
- 字符常量：单引号引起来，有且仅有一个字符，必须有内容。例如：'A'、'b'、' '(内容为空格)
- 字符串常量：双引号引起来，可以写多个字符，可以为空。例如："abc"、"Hello"、""(内容为空)
- 布尔常量：只有两个值，true、false
- 空常量：只有一个值null，代表没有任何数据

#.注意：

a.print、println的区别：

- System.out.print(); // print将他的参数显示在命令窗口，并将光标定位在所显示的最后一个字符之后
- System.out.println(); // println将他的参数显示在命令窗口，并在结尾加上换行符，将光标定位在下一行的开始

b.Java中常量的默认数据类型：整数常量默认为int、浮点数常量默认为double

四.变量和数据类型

1.数据类型

Java的数据类型分为两大类：基本数据类型和引用数据类型，并且只有这两大类。

(1).基本数据类型(四类八种)

四类	数据类型(八种)	关键字	内存占用	取值范围
整数型	字节型	byte	1个字节	-128~127
	短整型	short	2个字节	-32768~32767
	整型	int	4个字节	-2的31次方~2的31次方-1
	长整型	long	8个字节	-2的63次方~2的63次方-1
浮点型	单精度浮点型	float	4个字节	1.4013E-45~3.4028E+38
	双精度浮点型	double	8个字节	4.9E-324~1.7977E+308
字符型		char	2个字节	0-65535
布尔型		boolean	1个字节	true, false

(2).引用数据类型

字符串(String)、数组、类、接口、Lambda表达式等

#.注意：

a.byte、short、int、long、float、double的取值范围依次增大。由此可见数据范围与字节数不一定相关，例如float数据范围比long更广泛，但float是4字节，long是8字节。

b.双精度浮点型比单精度浮点型除了取值范围更大以外，其数据精度更高，即可表示的小数点后的位数更多。

2.变量

变量(Variable)：程序运行期间，内容可以发生改变的量。

- 先定义再赋值：　　数据类型　变量名称；　　变量名称 = 数据值;
- 定义时赋值：　　数据类型　变量名称 = 数据值;

#.注意：

a.没有进行赋值的变量，不能直接使用；一定要赋值之后，才能使用。

b.在同一个大括号范围内，如果创建多个变量，那么变量的名称不可以相同。

c.变量的作用域：从定义变量的一行开始，一直到直接所属的大括号结束为止。变量的使用不能超过其作用域的范围。

d.可以通过一个语句来定义和赋值多个变量，但是一般情况不推荐这么写。例如：int a, b, c; 或 int x=100, y=200, z=300;

e.定义long类型变量时，数据值一定要加上后缀L；定义float类型变量时，数据值一定要加上后缀F。如：long num=300L; float num=2.5F