

## 第一周学习笔记

07月16日

### 一、环境变量意义

1. 环境变量，指的是操作系统中一个具有特定名字的对象，它包含了一个或者多个应用程序所即将使用到的信息。以 path 变量为例，当我们运行一个程序的时候，如果没有告诉他程序所在的完整的路径的时候，系统就会在当下目录寻找这个程序，如果没有，系统就会进入 path 中指定的路径去寻找。因此我们通过设定环境变量，可以达到正确的运行进程的作用。
2. Java 环境变量配置的基本流程：
  - (1) JAVA\_HOME 设定 jdk 的安装目录，进而在设置 path 变量的时候只需要 % JAVA\_HOME %，相当于相对路径
  - (2) Path：指定 java 命令的搜索路径
  - (3) CLASSPATH：指定 java 中类包等文件的搜索命令，JVM 就是通过它来寻找类的。

### 二、java 命令行基本操作指令

- (1) javac：编译命令，将 java 源文件编译成.class 文件
- (2) java：运行字节码文件，jvm 对字节码进行编译和运行
- (3) where java 查看 jdk 安装位置
- (4) java -version 查看 jdk 版本

### 三、JDK JRE JVM 的区别和联系

1. JDK (Java development Kit), JRE (Java Runtime

Environment): jre 是运行 java 程序所必需的环境集合, 包括了 JVM 和 java 的核心类库在内, 而 jdk 是 java 语言的开发工具包, 一般 jre 会存在于 jdk 的目录下。

2. Jdk 下个目录的含义: bin 目录下放了 JDK 的各种工具命令 (.exe 文件); db 目录里面安装的是 java DB 的路径; include 目录下是一些供 C 语言使用的标题文件。C 语言的头文件支持 java 本地接口和 java 虚拟机调试程序接口的本地编程技术; jre 目录即为 Java 的运行环境; lib 目录下存放的是 jdk 工具命令的实际执行程序; src.zip 目录下存放的就是 java 核心类库的源代码; COPYRIGHT 和 LICENSE 为版权和签证说明文档

#### 四、 java 程序编写的基本流程和原理

1. 基本流程: 编写代码, javac 编译, java 运行字节码 (jvm 解释)

2. 原理:

(1) java 源代码 (.java 文件) 通过 Javac 编译为虚拟机可以运行的字节码文件 (.class 文件), 字节码文件再通过 jvm 虚拟机进行解释和运行

(2) .Java compiler (javac): 编译器首先扫描源代码生成首相语法树 (AST), 对其进行语法和词法的分析。第二步金星天蝎顾好表, 因为此时类的特定地址无法确定, 所以通过符号去代替。在装载类和接口的时候, jvm 用准确的地址去替换符号表。第三步进行注释处理, 第四步分析并且生成字节码。

(3) java 虚拟机 (jvm) 是介于机器和编译程序之间的一层虚拟的机器, 它为在不同的平台上为编译程序提供了一个相同的接口。编译好的程序只需要面对虚拟机, 生成虚拟机可以理解的代码, 然后

jvm 将编译程序解释为机器可以理解的代码（机器代码）

#### (4) JVM (Java Virtual Machine):

- (1) 每个 JVM 都包含两种机制：类装载子系统（装载适配的类或者接口），执行引擎（负责执行包含在类和接口的指令）。每个 JVM 都包含方法区，java 堆，java 栈，本地方法栈，指令计数器和其他的隐含寄存器。
- (2) Java 程序最早通过 jvm 中的 java 解释器进行解释执行，但是发现部分代码块或者方法多次出现，就会即将这些代码认定为热点代码，为了提高执行效率，运行时会将这些代码编译为与本地平台相关的机器码并且进行优化，这个过程有 JIT（即时编译器）执行。

### 五、class 文件时用来做什么的

- (1) java 类文件的属性方法以及类中的常量信息都会被分别存储在 class 文件中，并且可以被 jvm 识别
- (2) class 文件是跨平台的基础

### 六、git 是什么

git 是一个强大的分布式的版本迭代工具，并且可以通过网络将版本库推送到远程仓库。与集中式的 SVN，git 的执行速度特别快（指针），并且 git 记录的是修改而不是文件。git 分为工作区，暂存区，版本库三个部分。

### 七、git 常见操作命令

`git init` 初始化本地仓库 (生成.git); `git status` 查看仓库状态; `git add` 将文件添加到暂存区; `git commit -m “”` 将文件提交到版本库, 并且附带表述信息; `git diff` 查看修改内容; `git log` 查看提交日志而 `git reflog` 可以查看所有历史命令; `git reset --hard id` 回滚; `git checkout -- file` 丢弃工作的修改; `git reset HEAD` 将提交到暂存区的文件退回工作区; `git rm` 删除文件 ;

远程推送 (github):

- (1) github 与本地进行 ssh 关联 `git remote add origin`
- (2) git 上创建仓库
- (3) `git push -u origin master` 第一次推送到 master 分支, 后面若要修改就不需要 -u
- (4) `git clone` 克隆

## 八、对分支的理解

在 git 应该大力提倡使用分支, 主分支一般是已经经过测试并且功能齐全的的项目, 一般情况下不要在主分支进行修改 (特别容易造成分支冲突)。在不同的分支上进行最新功能的实现,

当其中两个分支的功能完成后, 可以进行分支合并, 这样一层的合并, 最后合并到主分支。

`Git branch` 查看分支; `git swith -c` 创建并且切换到新分支;

`Git swith` 切换分支; `git merge` 合并分支; `git branch -d` 删除分支

## 九、语法知识

### 1、Java 语言基础组成？（毕向东2-1）

关键字，标识符，注释，常量和变量，运算符，语句，函数，数组

### 2、关键字和标识符的区别？

关键字就是被 java 语言赋予了含义的单词，例如 `class` `byte` `static` 等；而标识符是我们用来为变量等一个实体起的名字，开头无数字，不用关键字，区分大小写

### 3、java 的基本数据类型：（毕2-7）

#### 基本数据类型（primitive type）

- 数值型
  - 整数类型（`byte` 1 字节、`short` 2、`int` 4、`long` 8）
  - 浮点类型（`float` 4、`double` 8）`float` 表示数值的范围比 `long` 还大
  - 字符型（`char` 2）必须要放一个字符或空格
    - Java 中所有的字符都是用 Unicode 编码
- 布尔型（`boolean`）

#### 引用数据类型（reference type）

- 类（`class`）
- 接口（`interface`）
- 数组（`[]`）

### 4、定义 `long` 类型的30，单精度的4.13，变量名为 `a`，（毕2-7）

```
long a = 30;
```

```
float = 4.13F;
```

### 5、定义数值4，字符4，字符串4

```
Int 4;char = "4";String = "4";
```

6、写出下面代码的运算结果：（毕2-9）

(1)  $a = 2 \% 5 \Rightarrow 2$ ;  $a1 = 2 \% -5 \Rightarrow 2$ ;  $a2 = -2 \% 5 \Rightarrow 3$ ;  
 $a3 = -2 \% -5 \Rightarrow -2$ ;

(2) `System.out.println("5+5="+5+5);`  $\Rightarrow$  "5+5" =55

7、定义 `short s = 5`；下面两句代码的区别。（毕2-11）

(1) `s = s + 3;`  $\Rightarrow$  编译不通过 3为 int 数值，会进行数值转换

(2) `s += 3;` 编译通过，不会改变 s 的数据类型

8、“&&”与“&”的区别？（毕2-12）

&&短路与，若符号前面的表达式不符合条件，结果直接为 false，不在进行的运算

&逻辑与，符号前后的表达式都会进行运算

“&”还可以做位运算符

9、“>>”和“>>>”的区别？（毕2-13）

“>>”为右移在一定范围内向右移动一位相当于除以2，第一位是0的数向右移动的空缺位补0，第一位是1的数向右移动的时候空缺位补1.

“>>>”无符号右移，最高位无论是0或者1都用0去填补空缺

10 交换两个变量的值。两种方式。（毕2-15）

(1) 采取中间变量

(2) 位运算

```
public class Test {  
  
    public static void main(String []args){  
  
        //1. 采取中间变量  
  
        int temp;  
  
        int a = 5;  
  
        int b = 3;  
  
        temp = a;  
  
        a = b;  
  
        b = temp;  
  
        System.out.println("交换后 a=" + a + ",b=" + b);  
  
        //2. 位运算  
  
        int x = 10, y = 20; //定义两个变量  
  
        x = x ^ y; //x = 30  
  
        y = x ^ y; //y = 10  
  
        x = x ^ y; //x = 20  
  
        System.out.println("交换后 x=" + x + ",y=" + y);  
  
    }  
  
}
```

11、for 和 while 的区别（毕3-4）

(1) for(初始化表达式;条件表达式;循环后的操作表达式) {循环体; }

(2) while(判断条件语句) {

    循环体语句;

    控制条件语句;}

(3)如果需要变量来对循环进行控制，该变量值作为循环增量存在的时候，有区别。For 再循环结束后将循环变量释放，节约内存空间

12、for 和 while 最简单的无限表达形式。(毕3-5)

For ( ; ; ) {}

While (true) {}

13、函数中形参和实参的区别。(毕3-12)

形参：是一个变量，用来储存调用函数的时候传递给函数的实际参数

实参：传递给形式参数的具体数值

14、优化下面的代码

```
for(int x = 0; x < arr.length; x++){  
  
    System.out.println(arr[x]); }
```



```
int x ;  
  
int size = arr.length;  
  
for(x = 0;x < size; x++){System.out.println(arr[x]);}
```