一:各环境变量的意义

1.环境变量的意义

操作系统中具有特定名字的对象，用于储存一个路径，该路径指明某些文件、程序的完整路径;JAVA 的环境变量便是指明 jdk、jrm 等文件的路径以便系统查找在编程过程中用到的相应的语法、函数所在的具体文件位置。

2.JAVA_HOME

储存 Java 的安装目录；听说有各规定就是当有软件需要用到 Java 时会直接在操作系统中找这个变量。

3.PATH

储存 Java 命令所在路径，即 jdk 下的 bin 目录所在路径，设置好 PATH 后可以在系统执行命令时使用 javac/java 等工具，所以 PATH 在 JAVA_HOME 之下。

4.CLASSPATH

储存 Java 的执行环境在哪些路径、目录之下，通常与 improt、package、class 有关，即 jdk 下的 lib，也在 JAVA_HOME 之下。

二:JDK、JRE、JVM 的区别

1.跨平台核心 JVM

Java 可以实现一处编写、多处运行，其原因就在于 JVM，在不同的的操作系统中 JVM 通过适应于本系统的独立的 JVM 实现将代码编译为唯一的字节码然后在 JavaAPI 层面调用相同的设备实现同一功能，也就是先得到同一的指令集(字节流)然后解释自己的指令集到 CPU 的指令集或 OS 的系统调用。

2.运行环境 JRE

包含底层的 JVM 实现(bin)、JVM 工作需要的基本类库(lib)，所以可以说是 Java 运行、开发时的环境。

3.开发工具 JDK

工具包，Java 开发的核心，包含编译器相关的文件(bin)、Java 与 JVM 交互的头文件(include)、类库(lib)、运行环境(jre)。

4.总结：Jdk 中有 jre，jre 中有 jvm，jvm 与操作系统相连。

三:流程

编写 Java 源文件
Javac 命令将其编译为 class 文件（字节码）
JVM 加载 class 文件

四:class 文件

Class 文件与操作系统无关所以可以在不同操作系统中使用为

Java 程序提供独立于底层的二进制形式服务。

五：Git

1.基本知识

主要是管理 GitHub，托管项目代码；
同时主要功能为管理代码的版本，便于用户查找以前版本；
本地有三个工作区域：
在工作区对文件修、添、改；
暂存区暂存修、添、改的文件方便再一次做进一步的修、添、改操作；
确认无误后将暂存区的文件提交到 git 本地仓库。

2.git 管理本地仓库

创建文件夹：mkdir 文件夹名
进入文件夹目录：cd 文件夹名/文件夹路径
初始化 Git 仓库（存储仓库所有信息）:git init
退到上一级：
当前文件状况：git status
创建文件：touch 文件名
删除文件：rm 文件名
修改文件内容：vi 文件名
查看文件内容：cat 文件名
将文件从工作区转到暂存区：git add 文件名
将暂存区中的文件删除：git rm 文件名
确认暂存区中执行的操作保存到仓库：git commit -m”描述”

3.git 管理远程仓库

从远程仓库中项目克隆到本地：git clone 仓库地址
在本地工作区对文件进行操作后添加到本地仓库的暂存区；
确认无误后确定添加到本地仓库；
将本地仓库同步到远程仓库：git push

六:对分支的理解

1.基本知识

分支是一次处理不同版本存储库的方法。
默认情况下，存储库有一个名为 master 的分支，该分支被视为最终分支。在将分支提交给 master 之前，我们使用分支进行实验和编辑。
当我在主分支上创建分支时，我正在按主分支在该时间点上的状态创建主分支的副本或快照。如果在我处理分支时其他人对主分支进行了更改，则可以引入这些更新。

七:语法知识

1.基本组成

关键字、运算符、标识符、语句、注释、函数、常量与变量、数组。

2.关键字和标识符

关键字：(Java 自带)

用于定义数据类型的关键字 class interface byte short int long float double char boolean void

用于定义数据类型值的关键字 true false null

用于定义流程控制的关键字 if else switch case default while do for break continue return

用于定义访问权限修饰符的关键字 private protected public

用于定义类，函数，变量修饰符的关键字 abstract final static synchronized

用于定义类与类之间关系的关键字 extend simplements

用于定义建立实例及引用实例，判断实例的关键字 new this super instanceof

用于异常处理的关键字 try catch finally throw throws

用于包的关键字 package import

其他修饰符关键字 native strictfp transient volatile assert

标识符：(自定义)

遵循大小写严格、不能与关键字冲突、不能以数字开头

3.基本数据类型

数值型：byte short int long float double

字符型：char

布尔型：boolean

4.引用数据类型

类 class、接口 interface、数组[]

三:简答题

1、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 a = 4.13;
```

2、定义数值 4，字符 4，字符串 4

```
Int a = 4;
Char a = ‘4’;
String a = “4”;
```

3、写出下面代码的运算结果：(毕 2-9)

(1)a = 2 % 5; a1 = 2 % -5; a2 = -2 % 5; a3 = -2 % -5;

2;2;-2;-2

(2)System.out.println("5+5="+5+5);
5+555

4、定义 short s = 5; 下面两句代码的区别。(毕 2-11)
(1) s = s + 3;
S+3 的结果会转化自动提升到 int 型，而 short<int 数据有可能溢出，因此会报错。
(2) s += 3;
+=后面的数据在进行运算前会强制转换为 s 对应的数据类型，因此不会报错

5、"&&"与"&"的区别？(毕 2-12)
前者仅当左边为真时右边才参与运算;
后者无论左边真假右边都参与运算

6、">>"和">>>"的区别？(毕 2-13)
>>运算中被移位的二进制最高位是 0，右移后，空缺位补 0;最高位是 1，空缺位补 1。
>>>被移位二进制最高位无论是 0 或者是 1，空缺位都用 0 补。

7、交换两个变量的值。两种方式。(毕 2-15)
利用第三变量做中间值:
```
Int x,a,y;
a=x;x=y;y=a;
```
通过位运算:
```
int x,y;
x=x^y;y=x^y;x=x^y;
```

8、for 和 while 的区别 (毕 3-4)
在实现的功能上可以互化，但 for 的循环变量在 for 结束后会被释放内存。而 while 循环使用的变量在循环结束后还可以继续使用。

9、for 和 while 最简单的无限表达形式。(毕 3-5)
```
while(true);
for(;;);
```

10、函数中形参和实参的区别。(毕 3-12)
形参: 函数里面，由实参赋值，函数结束释放内存;
实参: 函数外面，为形参赋值，不随函数发生变化，可由函数返回值重新赋值;

11、优化下面的代码
```
for(int x = 0; x < arr.length; x++){
    System.out.println(arr[x]); }

int y=arr.length;
for(int x = y; x < y; x--){
```

```
System.out.println(arr[x]); }
```