1. 设置Java开发环境；

安装完成以后需要添加环境变量:

Windows:

1,java\_home:java的安装目录

2,classpath : .

3,path :%java\_home%/bin;

Linux:

配置环境变量(文件: .bashrc 这文件的最后一行开始配置)

export JAVA\_HOME=/home/wood/java/jdk1.7.0\_79

export CLASSPATH=.:$JAVA\_HOME/lib:$CLASSPATH

export PATH=$JAVA\_HOME/bin:$PATH

配置完了之后,需求让文件立刻生效: source .profile

mac : 请参考 https://www.cnblogs.com/dingzhijie/p/7016397.html

安装完成以后 在cmd中输入 java 和javac 有反应即视为安装成功

2. 理解Java语言特征；

Java语言包括语法规则和语法规则的实现。Sun公司对Java语言有自己的一份参考实现：JDK。

a. 编程语言：语言本质即规则: 字符集、词法规则、句法规则(条件语句、循环语句等);

桌面应用程序 拥有图形化用户接口的软件；

服务器应用程序：通过浏览器访问运行；

b. 开发环境：开发程序，不管是桌面应用还是服务器应用，均需要JDK的支持。

c. 应用环境：运行时需要JRE的支持。

为什么java需要jre呢，因为java需要跨平台，也就是在什么机器上都能运行良好！

java源文件会被编译成一个中间文件.class文件，这个文件不能像.exe文件那样直接运行，而是被jre翻译成当

前机器运行时需要的程序，之后在运行。所以没有jre，那么.class文件就无法转换成可运行的程序，所以脱离jre，

java不可能运行

d. 部署环境：服务器应用程序需部署运行在服务器应用程序上，服务器本身也是一个Java程序

用Java语言编写的服务器应用程序部署运行时也需要Java语言的支持；

3. 为什么发明Java?

为什么发明了java，换而言之，java有哪些优点呢？

a. 提供一个解释环境

一次编译到处运行: 跨平台；

多线程：多线程编程的简单性是Java成为流行的服务器端开发语言的主要原因之一

b. 提供一个比较容易的方式编程

1) 更健壮：没有指针(引用)、没有内存管理；

2) 纯粹的面向对象的编程：在Java中认为一切均是对象，对象有属性以及改变属性值的方法。通过Java编程围绕

着构建对象的模板、实例化对象、调用对象的方法和属性等来进行。

封装 继承 多态

c. 那么如何实现上面这些呢？

虚拟机JVM提供一个解释环境，让java代码可以运行。

垃圾回收机制让程序员无需在代码中进行内存管理。

代码安全校验保证java代码的安全性。

4. 理解Java虚拟机的特征；

Jvm是什么？简称java 虚拟机（java virtual machine），是用软件来模拟一个虚拟的环境。

我们用java编写的源程序经过编译以后生成字节码文件，Jvm 提供了一个解释运行 java字节码文件的环境，

只要在不同操作系统上安装了jvm后，就能对同一个java程序进行解释运行，这就java的跨平台，一次编译，到处运行。

将生成.class转换成机器或者(操作系统)所识别的二进制文件

5. 描述垃圾收集的概念；

什么叫垃圾回收：

1) 垃圾：无用对象所占据的内存空间；

垃圾回收：将无用对象所占据内存空间进行回收的过程；

为什么要垃圾回收？

1) 对象创建，需要占用内存空间，在一个程序运行过程中要创建无数个对象，如果对象永久占用内存，那么内存很快会

消费光，导致后续对象无法创建或者出现内存不足的错误。因此必须采取一定措施及时回收那些无用对象的内存，这

样保证内存可以被重复利用。

C++等程序由程序员显示进行内存释放，这样有可能：

1) 忘记内存释放，导致无端的内存占用；

2) 释放核心类库占用内存，导致系统崩溃；

Java中垃圾回收处理：

1) 由虚拟机通过一个系统级的垃圾回收器线程自动完成，不会忘记也不会释放错，系统更加稳定；

Java中垃圾回收处理特点：

1) 由虚拟机通过垃圾回收器线程自动完成；

2) 只有当对象不再被使用，它的内存才有可能被回收；如果虚拟机认为系统不需要额外的内存，即便对象不再使用，内存

也不会回收；

cpu是否空闲,内存是否紧张

3) 程序无法显示迫使垃圾回收器立即执行垃圾回收，可以通过java.lang.System.gc()/java.lang.Runtime.gc()建议

虚拟机回收对象；

6. 定义类、包、applets和应用程序；

1) Java的工作方式：

源代码: 编写Java源代码文件

|

编译器: 用编译器编译源代码。编译器会检查错误，如果有错就要改正才能产生正确的输出。

|

输出: 编译器会产生字节码。任何支持Java的装置都能够把它转译成可执行的内容。编译后的字节码与平台无关。

|

Java虚拟机: 可以读取与执行字节码。加载指定类，开始执行它的main方法，然后一直运行到main的所有程序代码

结束为止。

2) Java的程序结构：

a. 源文件：

包含Java代码(按java语法规则编写的内容)的文件;

特征：

i. 以java作为后缀；

ii. 可以包含多个类/接口；

iii.名称与类名(只包含一个类/接口)或public修饰的类/接口(包含多个类/接口)同名;

结构：

i. 包的声明语句(可有可无，如有只能有一行，且位于最前面);

ii. import语句(可有可无, 可有多行，紧跟着包的声明语句);

iii.类的声明语句{};

b. 类：

Java代码组织的单位, Java代码都是以一个个类形式组织的。用Java语言编写程序其实就是编写一个个类；

一个类里的语句内容用大括号括起来，一个类里带有零至多个方法。

结构：

i. 类的声明语句，例：

public class HelloWorld

ii. 类的内容语句，用大括号括起来：

(1) 属性；

(2) 方法：

c. 方法：

类似于其它语言里的函数或过程。方法里带有零至多行语句。

结构：

i. 方法的声明语句，例：

public static void main(String[] args)

ii. 方法体，用大括号括起来：

(1) 属性；

(2) Java语句;

d. Main方法：

Java虚拟机执行Java程序时，首先执行调用的方法。main方法又叫程序的入口方法或起点方法。不管你的程序有多大，

不管你的程序有多少个类，一定都会有一个main()方法来作为程序的起点。

3) 剖析类：

public class FirstJavaProgram{

public static void main(String[] args) {

System.out.println("Hello Java, I am your fans!");

}

}

执行

javac HelloWorld.java

如果没有报错,表示编译通过

如果编译通过,执行java HelloWorld 运行helloworld程序

注意事项:

1.允许构建的类中不存在public修饰的class,如果没有public

修饰的类,此时在其中构建main方法.也能作为程序入口

public修饰的类必须和当前文件名称一致(大小写必须一致)

helloworld.java public class HelloWorld 报错).

2.一个java文件中只允许有一个public修饰的类,并且该类的名称

必须和当前文件名称一直.类的名称与文件必须大小写一致.

允许类名称小写,只要和文件名称一致.允许有多个普通类

3.public static void main(String args[]){} 程序的入口所有的方法的构建

public去掉 编译通过,运行时发现找不到main方法.程序不能运行

static去掉 编译通过,运行时发现找不到main方法.程序不能运行

void去掉 编译不通过 报方法缺少返回类型

public和static位于可以置换,因为都是修饰符.

其他内容的位置都不能置换,必须满足方法构建的顺序

修饰符 ＋ 返回类型 ＋ 方法名(参数类型 参数名称…){}

main名称不一致, 编译通过,运行时发现找不到main方法,程序不能运行

main可以和后面()存在空格,jvm仍然理解成main方法

String类型发生改变,编译通过,运行时发现找不到main方法,程序不能运行

String类型大小写敏感.不能随意更改,编译不通过.

args名称可以发生改变,jvm仍然理解成main方法

[]位置可以发生改变,可以位于String后面或者args后面,但是不能不写

如果不写,编译通过.jvm找不到main方法

main方法在一个类中只能出现一次

4.package import public class 的顺序

package最先并且只能有一个,但是可以不位于第一行

import必须紧跟package,可以有多个

java.lang包是默认导入的jar包,换言之这个包下面的所有类默认被导入.

public class必须位于package以及import之后,但是和其他class没有顺序

javac:将.java文件生产.class文件.如果.java中包含package

-d .选项可以将.java文件中制定的package内容生成对应

目录.e.g HelloWorld.java—>package com.briup.chap01;

-d . 可以在当前目录下生成com目录,com目录中在生成briup目录

briup目录中再生成chap01目录.此时chap01目录中包含生成好的

.class文件如果-d bin选项,不在当前目录下生成对应的package制定目录.

而是在bin目录下生成对应的package制定目录

src目录存放.java文件

bin目录存放.class文件

7.重要的java命令

1编译java源文件

Javac <options> <source file>

2,执行类

Java [-options] class [args1 args2 args3 ...]

3,创建java文档

Javadoc [option] [packagenames] [sourcefiles]

4,生成jar包

jar {ctxu} [jar-file] [-C directory] files ...

8. 注释

类型：1) 单行注释：

//text——从"//"到本行结束的所有字符均作为注释而被编译器忽略

2) 多行注释：

/\*text\*/——从"/\*"到"\*/"间的所有字符会被编译器忽略

/\*\*/多行注释，可以出现在任意位置(包括[]的内部),

但是不能出现在关键字的中间 e.g main—>ma/\*test\*/in 这是不允许的

3) 文档注释：

/\*\* text \*/——从"/\*\*"到"\*/"间的所有字符会被编译器忽略。当这类注释出现在任何声明(如类的声明、类的成员变量的声明或者类的成员方法的声明)之前时，会作为JavaDoc文档的内容；

9. 分号、块和, 标识符

1) 每个语句短语以;结束 ;

2) 代码段以{}结束;所有的逻辑语句必须出现在{}中间

3) 空白处（空格、tab键、新行和回车（几个语句短语连接在一起））是无关紧要的。

4) 类、方法和变量的名字

java标识符以字母、\_和$开头，不能以数字开头，后跟字母、数字、“\_”和“$”的组合;

大小写敏感

没有长度限制。

举例： 合法标识符 非法标识符

try try#

GROUP\_7 7GROUP

openDoor open-door

boolean1 boolean

10.关键字

Java语言的关键字是程序代码中的特殊字符。包括：

. 类和接口的声明——class, extends, implements(实现), interface

. 包引入和包声明——import, package

. 数据类型——boolean布尔, byte, char, double, float, int, long, short

. 某些数据类型的可选值——false, true, null

. 流程控制——break, case, continue, default, do, else, for, if, return, switch, while

. 异常处理——catch, finally, throw, throws, try

. 修饰符——abstract, final, native, private, protected, public, static,

synchronized(同步锁), transient

. 操作符——instanceof 比较类型是否一致

. 创建对象——new

. 引用——this引用当前对象, super引用父类对象

. 方法返回类型——void 没有返回类型