Java语言基础 Day01

- 1. 构建Java开发环境
- 2. Java HelloWorld
- 3. 基于Eclipse的JavaHelloWorld

1 构建Java开发环境

1.1 问题

安装JDK,配置环境变量。

1.2 方案

学习Java语言要从Java SE平台开始。Oracle官方提供了两种针对Java SE平台的产品——JRE 和JDK,可以从官方网站免费下载(http://www.oracle.com/technetwork/java/index.html)。JRE(Java SE Runtime Environment)称之为Java SE运行时环境,提供了运行Java应用程序所必须的软件环境,包含有Java虚拟机(JVM)和丰富的类库(Libraries)。无论是开发Java应用还是仅仅运行一个已经开发好的Java应用都必须安装JRE。JDK(Java Development Kit)称为Java开发工具包,是JRE的超集,或者说JDK包含了JRE。JDK中除了包含有JRE的所有内容之外还提供了编写Java程序所必须的编译器和调试工具等。对于编写Java开发的人士一般需要下载JDK,目前的主流版本为JDK 6。

Java官方提供了针对不同操作系统平台的JDK版本,如Windows、Linux、Solaris等,在下载JDK时,可以根据自己的需要来选择,如图-1所示:

Java SE Development Kit 6u24



图-1

以Windows版本的JDK为例,下载完成后需要进行安装。和安装其他Windows下的软件一样,根据安装界面的提示(如图-2所示)选择好安装目录,然后下一步、下一步、...



图-2

需要注意的是,在图-2的JDK安装功能列表中Java DB是可以选择不安装的,这是一个JDK自带的纯Java语言实现的关系型数据库,通常的Java开发时用不到它的。另外,对于Windows版的JDK安装程序,除了安装一个已经包含有JRE的JDK之外,还要再安装一个独立的JRE(所谓"公共的JRE")并在注册表中对其进行注册,其目的是对于一些需要用到JRE的应用程序可以通过注册信息自动的找到JRE。"公共JRE"是一个可选项,可以选择不安装,毕竟JDK中已经包含有了一个完整的JRE(只不过没有在注册表中注册);如果选择了该项,在安装完JDK之后,安装程序还要提示用户选择公共JRE的安装目录。

除了下载JDK之外,开发人员一般还要下载Java官方文档,这是学习和开发Java语言必备的资料,具体下载位置在

http://www.oracle.com/technetwork/java/javase/downloads/index.html 页面的如图-3所示的位置。

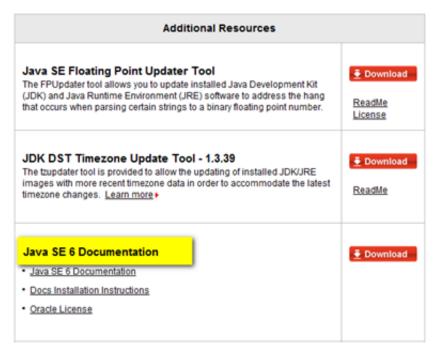


图-3

文档下载解压后以HTML格式呈现,使用浏览器打开文档首页——index.html,可以看到如图-4所示的"JDK全貌",通过单击相关链接可以了解其细节。

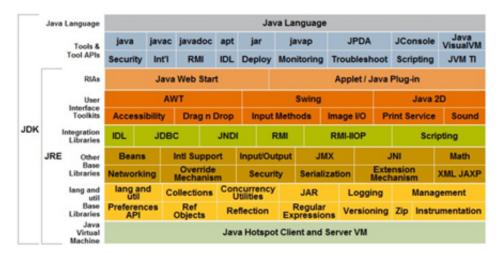


图-4

在图-4中可以看出: JDK包含JRE和开发工具包(Tools & Tool APIs);而JRE包含有Java虚拟机(JVM)和各种类库(Libraries)。

JDK安装完成后,在JDK安装路径下可以看到如下内容:

bin目录:用于存放JDK工具命令,比如用于编译Java程序的javac命令;用于启动JVM运行 Java程序的java命令;用于生成文档的javadoc命令;用于打包的jar命令等等。

jre目录:用于存放JDK所包含的JRE,其中包含有JVM和核心类库。

lib目录:用于存放JDK工具命令所对应的工具包(Tool APIs)。

demo目录:用于存放一些示例程序。

src.zip文件:用于存放核心类库的Java源代码。

其中bin中的javac命令和java命令是我们很快要用到的命令。如果想用到这些命令,需要让操作系统可以找到这些命令文件所在的路径。在Windows操作系统中,可以通过配置Path环境变量来实现。Path环境变量是一串用分号(;)分隔开的目录,在通过控制台运行一个命令时,Windows会依次在Path环境变量中的每一个目录中查找该命令,如果找到就可以执行,否则就会有如图-5所示的错误提示:



图-5

在Windows系统中通过"控制面板"→"系统"→"高级系统设置"打开如图-6所示对话框



图-6

单击"环境变量"按钮在弹出的对话框中双击系统变量Path,并对其进行编辑。在Path变量值的末尾追加JDK开发工具路径("JDK安装路径\bin"),假设JDK的安装路径为:"D:\Program Files\Java\jdk1.6.0_24"则需追加的Path路径为:";D:\Program Files\Java\jdk1.6.0_24\bin"。注意,路径之间需要用分号隔开。如图-7所示。

变量	值(-	
PATH	D:\Program	编辑系统变量			Σ
TEMP TMP	%USERPROFI %USERPROFI	变量名(8):	Path		
	新建 (N)	变量值 (V):	\Program	Files\Java\jdk1.(6. 0_24\Ъі
系统变量 (S)				确定	取消
空量	16		_		
	值 Windows NT				
0S	Windows_NT	system32;C:\Win			
变量 OS Path PATHEXT	Windows_NT C:\Windows\	system32;C:\Win BAT;.CMD;.VBS;.\	dows;		

图-7

环境变量设置完成后,通过"附件"→"命令提示符"(或者运行cmd命令)打开控制台,键入java或javac命令,看到输出正常的提示信息就表示环境变量配置成功了。如图-8示。

```
_ 0
                                                                     23
面 管理员: C:\Windows\system32\cmd.exe
C:\Users\electron>javac
用法: javac 〈选项〉〈源文件〉
其中,可能的选项包括:
                         生成所有调试信息
不生成任何调试信息
 -g:none
                          只生成某些调试信息
  g: {lines, vars, source}
                         不生成任何警告
  -nowarn
                         输出有关编译器正在执行的操作的消息
 -verbose
                         输出使用已过时的 API 的源位置
 -deprecation
                           指定用已过时的 API 的源位立
指定查找用户类文件和注释处理程序的位置
指定查找用户类文件和注释处理程序的位置
指定查找输入源文件的位置
覆盖引导类文件的位置
覆盖等类的扩展目录的位置
 -classpath <路径>
 -cp <路径>
  -sourcepath 〈路径〉
 -bootclasspath <路径>
 -extdirs 〈目录〉
                           覆盖签名的标准路径的位置
 -endorseddirs <目录>
 -proc: {none, only}
                         控制是否执行注释处理和/或编译。
 -processor <class1>[,<class2>,<class3>...]要运行的注释处理程序的名称;绕过默认
的搜索进程
                           指定查找注释处理程序的位置
 -processorpath <路径>
```

图-8

需要提及一点的是,如果在安装JDK是选择安装了公共JRE,则不需要配置Path环境变量也可以运行java命令,这是由于公共JRE路径写入了注册表的缘故。

2 Java HelloWorld

2.1 问题

使用记事本编写HelloWorld.java程序,在控制台输出"Hello World"。运行界面如图-9所示:

```
D: \>javac HelloWorld.java
D: \>java HelloWorld
HelloWorld
```

图-9

2.2 方案

使用记事本建立一个名称为HelloWorld.java的文本文件。定义一个类,代码如下所示:

01. public class HelloWorld {}

定义Java应用程序的入口方法main,代码如下所示:

01. public static void main(String[] args) {}

在控制台打印输出"Hello World",代码如下所示:

```
01. System. out. println("Hello World");
```

2.3 实现

系统代码实现如下:

```
01. public class HelloWorld {
02. public static void main(String[] args) {
03. System. out. println("Hello World");
04. }
05. }
```

2.4 扩展

使用记事本编写HelloJava.java程序,在控制台输出一行"Hello",再输出一行"Java"。运行界面如图-10所示:



图-10

系统代码实现如下:

```
01. public class HelloJava {
02. public static void main(String[] args) {
03. System. out. println("Hello");
04. System. out. println("Java");
05. }
06. }
```

3 基于Eclipse的JavaHelloWorld

3.1 问题

使用Eclipse开发工具编写HelloWorld.java程序。提示用户在控制台输入她的姓名,系统把用

户的姓名和",欢迎你进入Java的世界",连接成一个字符串并输出到控制台。系统交互界面如图-11所示:

```
■ Console 

<terminated > HelloWorld [Java Application] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe 请输入你的姓名:
jessica
jessica, 欢迎你进入Java的世界
```

图-11

3.2 方案

使用Scanner类来实现交互,接收用户在控制台的输入,Scanner类的next()方法返回从控制台接收到字符串。代码如下所示:

```
01. Scanner scanner = new Scanner(System. in);
02. System. out. println("请输入你的姓名: ");
03. String name = scanner. next();
```

把接收到的用户输入的信息和",欢迎你进入Java的世界"拼接成一个字符串输出到控制台。 代码如下所示:

```
01. System. out. println(name + ", 欢迎你进入Java的世界");
```

3.3 实现

系统代码实现如下:

```
01. public class HelloWorld {
02. public static void main(String[] args) {
03. Scanner scanner = new Scanner(System. in);
04. System. out. println("请输入你的姓名: ");
05. String name = scanner. next();
06. System. out. println(name + ", 欢迎你进入Java的世界");
07. }
08. }
```

3.4 扩展 <u>Top</u>

把用户的姓名、",欢迎你加入"和班号连接成一个字符串并输出到控制台。系统交互界面如图-12所示:

```
■ Console 

<a href="text-align: left) | Console |
```

图-12

系统代码实现如下:

```
01.
       public class HelloJava {
02.
           public static void main(String[] args) {
               Scanner scanner = new Scanner(System.in);
03.
04.
               System. out. println("请输入你的姓名:");
05.
               String name = scanner.next();
               System. out. println("请输入你的班号:");
06.
               String className = scanner.next();
07.
08.
               scanner. close();
               System. out. println(name + ", 欢迎你加入" + className);
09.
10.
11.
                                                                          隐藏
```