

Java开发环境搭建

丁勇

Email: 18442056@QQ.com

Java语言简介

Java开发环境构建

第一个Java程序

Java语言简介

➤ Java语言特点

Java是一个广泛使用的网络编程语言，它简单、面向对象，不依赖于机器的结构，不受CPU和环境的限制，具有可移植性、安全性，并且提供了并发的机制，具有很高的性能。此外，Java还提供了丰富的类库，使程序设计人员可以方便地建立自己的系统。

(1) 简单性。Java语言是一种面向对象的语言，语法规则和C类似，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java略去了指针、运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集机制大大简化了程序设计师的内存管理工作。另外，Java对环境要求非常低，它的基本解释器及类的支持只有几十KB，加上标准类库和线程的支持也大约200KB。

(2) 面向对象。Java语言的设计集中于对象及其接口，它提供了简单的类机制及动态的接口模型。对象中封装了它的状态变量及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法实现代码的复用。

Java语言特点

(3) 分布性。Java是面向网络的语言，通过它提供的类库可以处理TCP/IP协议，用户可以通过URL地址在网络上很方便地访问其他对象。

(4) 安全性。用于网络、分布环境下的Java必须要防止病毒的入侵。Java不支持指针一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

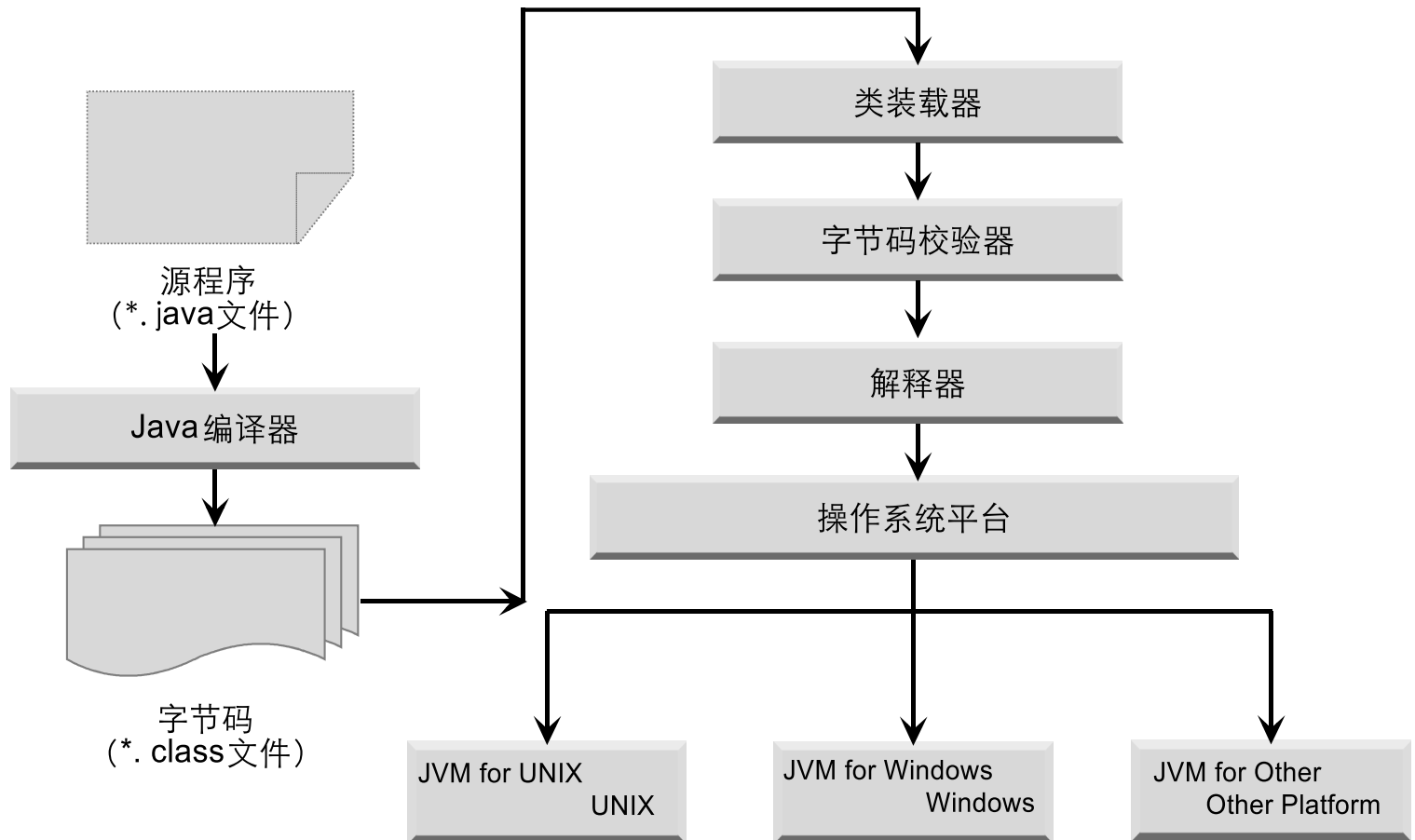
(5) 可移植性。与平台无关的特性使Java程序可以方便地被移植到网络上的不同机器。同时，Java的类库中也实现了与不同平台的接口，使这些类库可以移植。

(6) 高性能。和其他解释执行的语言不同，Java字节码的设计使之能很容易地直接转换成对应于特定CPU的机器码，从而得到较高的性能。

(7) 多线程。多线程机制使应用程序能够并发执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易地实现网络上的实时交互行为。

Java运行机制

Java程序是如何做到“一次编译，到处运行”呢？这正是通过Java虚拟机来实现的。JVM可以理解成一个以字节码为机器指令的CPU。首先，Java编译程序将后缀名为.java的Java源程序编译为JVM可执行的代码，即后缀名为.class的Java字节码文件。



Java运行机制

垃圾收集机制

垃圾回收器能够自动回收垃圾，即无用的对象所占据的内存空间被回收。在C/C++语言中，这些工作由程序员负责，无疑增加了程序员的负担。而Java语言消除了程序员回收垃圾的责任：它提供一种系统级线程来跟踪存储空间的分配情况，并在JVM的空闲时，检查并释放那些可被释放的存储空间。在Java中，对象被创建后，就会在堆区中分配一块内存。当对象不再被程序引用时，它就变成一个垃圾，所占用的堆空间可以被回收，以便空间被后续的新对象所使用。Java的垃圾回收器能断定哪些对象不再被引用，并且能够把它们所占据的堆空间释放出来。

Java语言简介

Java开发环境构建

第一个Java程序

Java开发环境的搭建

要想编译和运行上面的程序，离不开Java的编译和运行环境。SUN公司提供了自己的一套Java开发环境，通常称之为JDK(Java Development Kit)，又称J2SDK。目前最新的版本是Java SE Development Kit (JDK) 6 Update 10，可以到SUN公司的网站下载。输入网址<http://java.sun.com/javase/downloads/index.jsp>，进入第二栏，单击“Download”按钮。

Java SE Development Kit (JDK) 6 Update 10

The Java SE Development Kit (JDK) includes the Java SE Runtime Environment (JRE) and command-line development tools that are useful for developing applets and applications.

» Download



» More info about Java SE 6 Update 10 ...

[FAQ](#) | [Installation Instructions](#) | [ReadMe](#) | [ReleaseNotes](#) | [Sun License](#) | [Third Party Licenses](#)

Java开发环境的搭建

进入下图，在“Platform”栏中选择Windows，如果自己的平台是Linux操作系统，请选择Linux。选上前面的按钮，单击“Continue”按钮。

Select Platform and Language for your download:

Platform:

Language:

☒ I agree to the [Java SE Development Kit 6 License Agreement](#)

Continue »

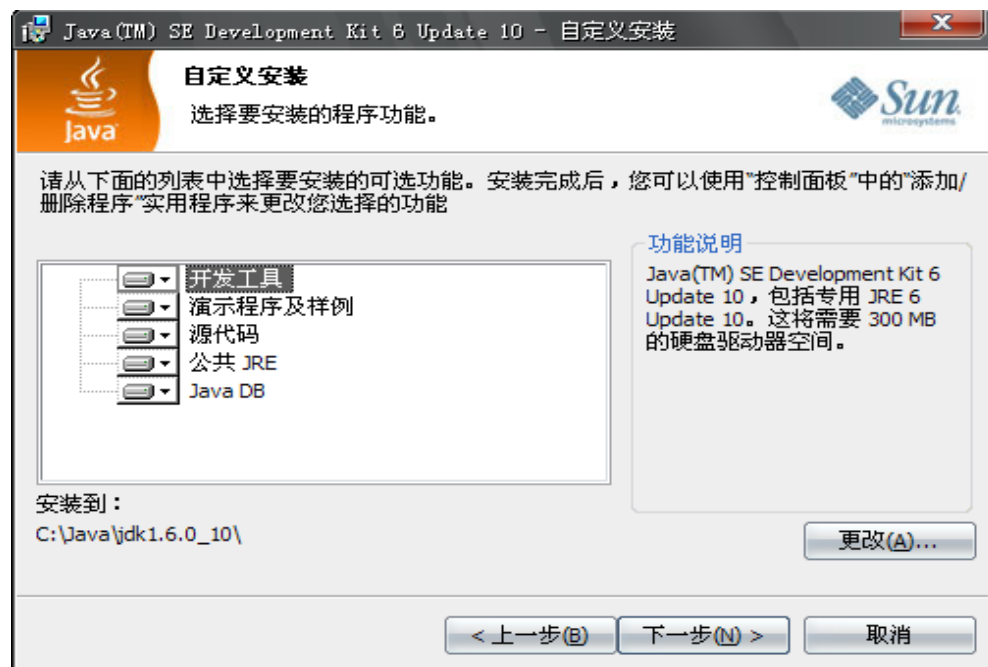
之后进入下图，在单选按钮上打钩，JDK就可以开始下载了。

Required Files

<input checked="" type="checkbox"/>	File Description and Name	Size
<input checked="" type="checkbox"/>	Windows Offline Installation jdk-6u10-windows-i586-p.exe	72.88 MB

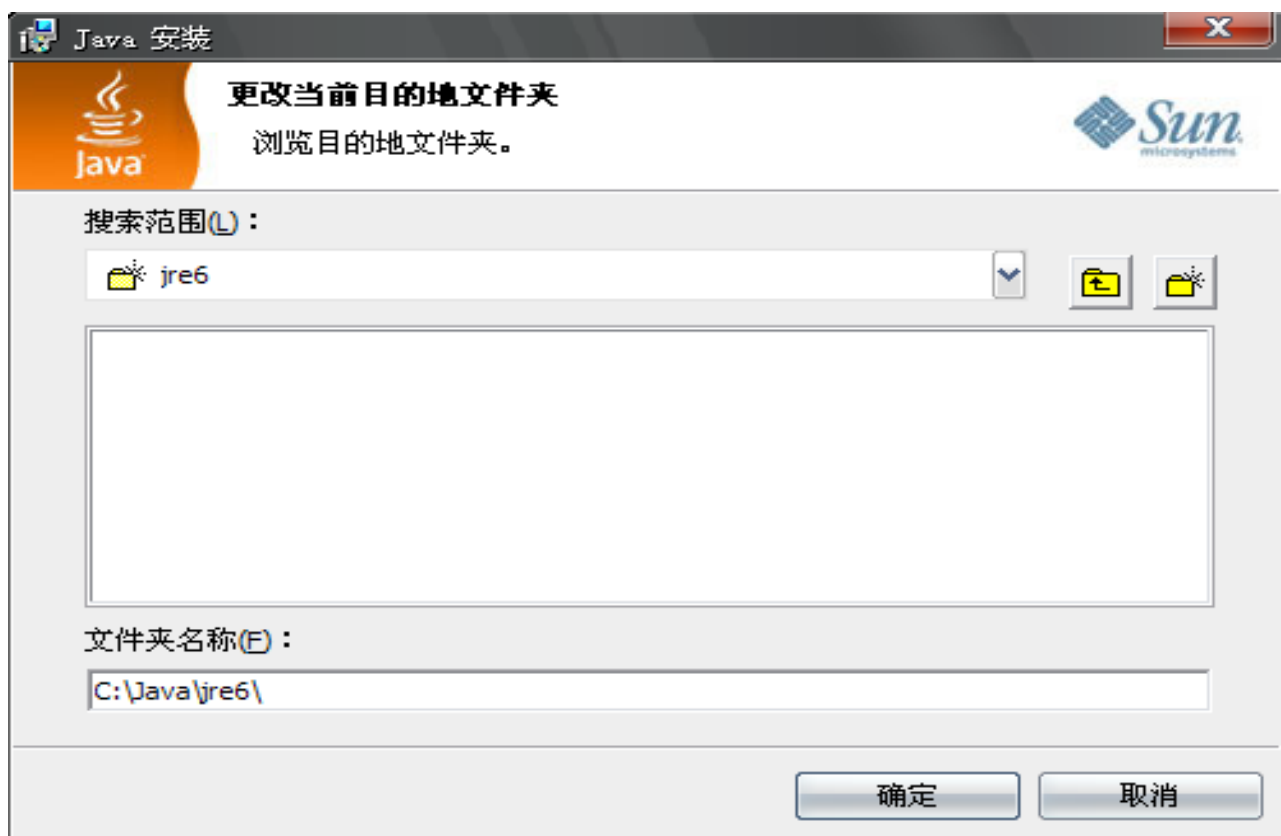
Java开发环境的搭建

下载完成后，双击可执行文件jdk-6u10-windows-i586-p.exe。按照提示完成安装，这里JDK的安装路径改为“C:\Java\jdk1.6.0_10\”



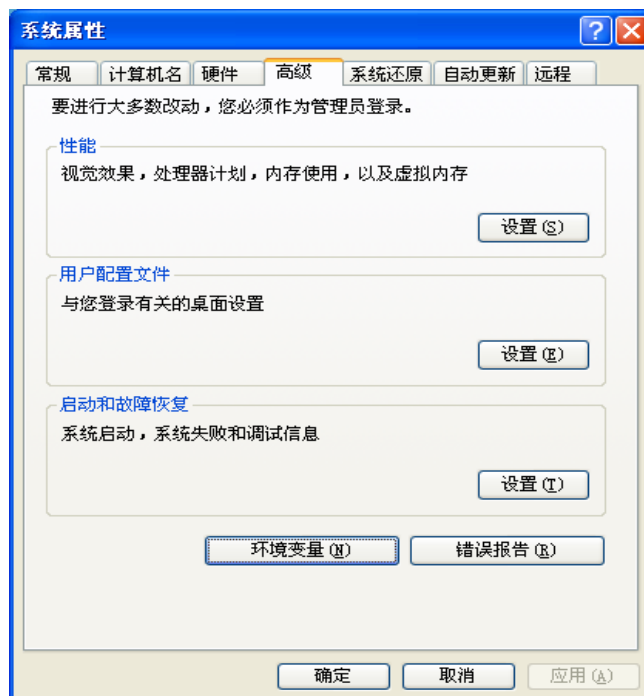
Java开发环境的搭建

在JDK的安装包里带有JRE安装包，JRE是Java的运行时环境。JRE的安装路径改为“C:\Java\jre6\”



Java开发环境的搭建

安装完成之后，还需要配置环境变量。在Windows 系统中，在桌面上右击“我的电脑”，从打开的快捷菜单中选择“属性”命令，在打开的“系统属性”对话框中单击“高级”选项卡；单击“环境变量”按钮，打开“环境变量”对话框



“系统属性”对话框



“环境变量”对话框

Java开发环境的搭建

进入JDK的安装路径，将看到如表所示的目录。

目录	描述
src.zip子目录	存放Java提供的API类的源代码压缩文件
bin子目录	存放JDK的工具程序
lib子目录	存放JAR文件，执行工具程序时，调用lib目录中对应的类
demo子目录	存放Java示范程序
jre子目录	JDK自己附带的jre，开发程序时做测试之用
db子目录	JDK6 中 附 带 的 Apache Derby 数 据 库， 支持 JDBC4.0

Java开发环境的搭建

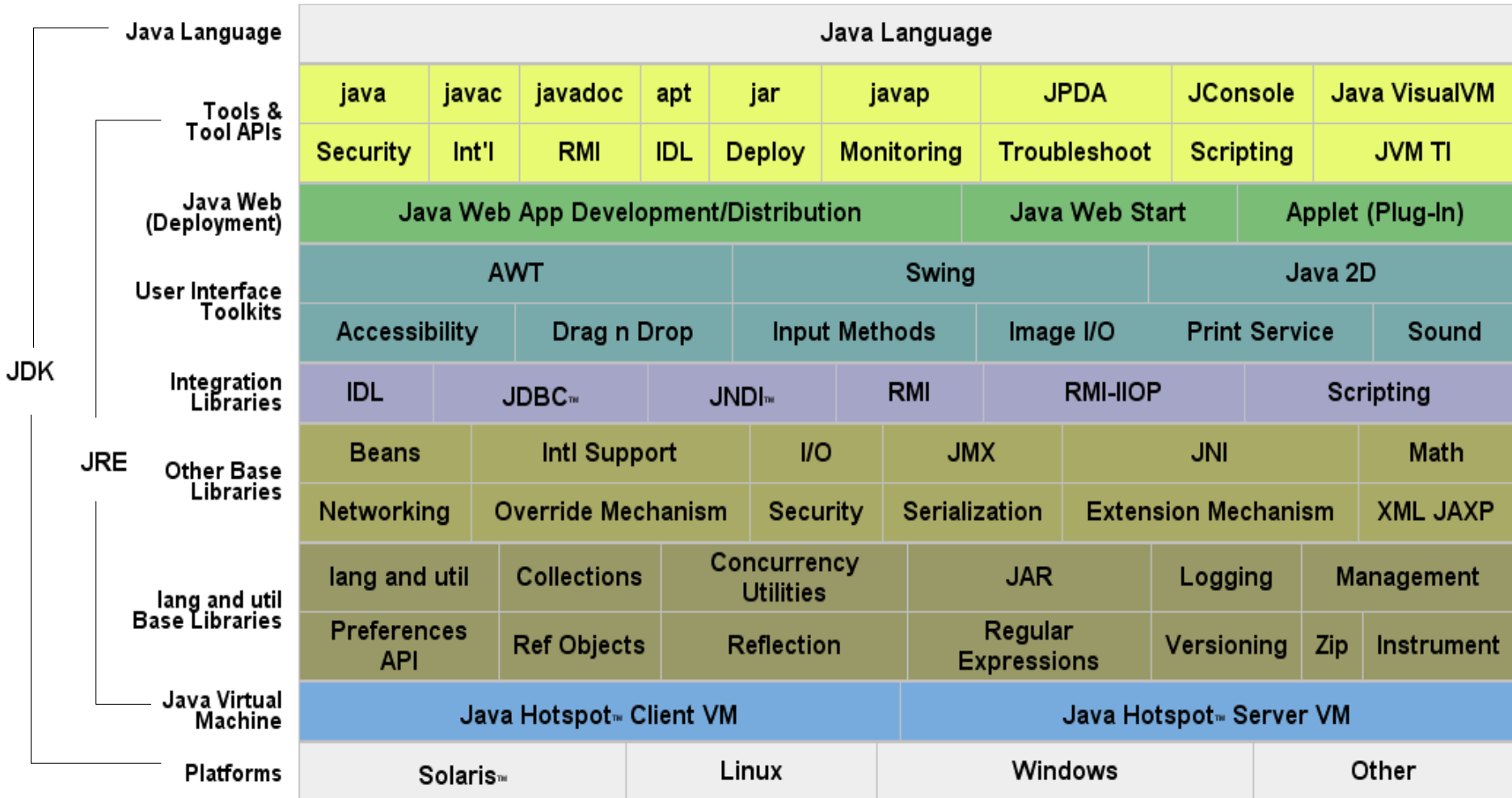


图 Java™ Platform, Standard Edition 6 JDK

Java开发环境的搭建

JDK主要包括以下内容：

(1) Java虚拟机：负责解释和执行Java程序，Java虚拟机可以运行在各种操作系统平台上。

(2) JDK类库：提供了最基础的Java类库即各种实用类，包括最常用的类库，java.lang、java.io、java.util、javax.swing和java.sql等。

(3) 开发工具：这些开发工具都是可执行程序，主要包括javac.exe(编译工具)、java.exe(运行工具)、javadoc.exe(生成Javadoc文档的工具)和jar.exe(打包工具)等。

Java的环境变量已配置好，现在就可以编写Java程序了。用记事本就可以编写Java程序。首先建立一个名为“Area.java”的源文件，存放在d盘根目录。

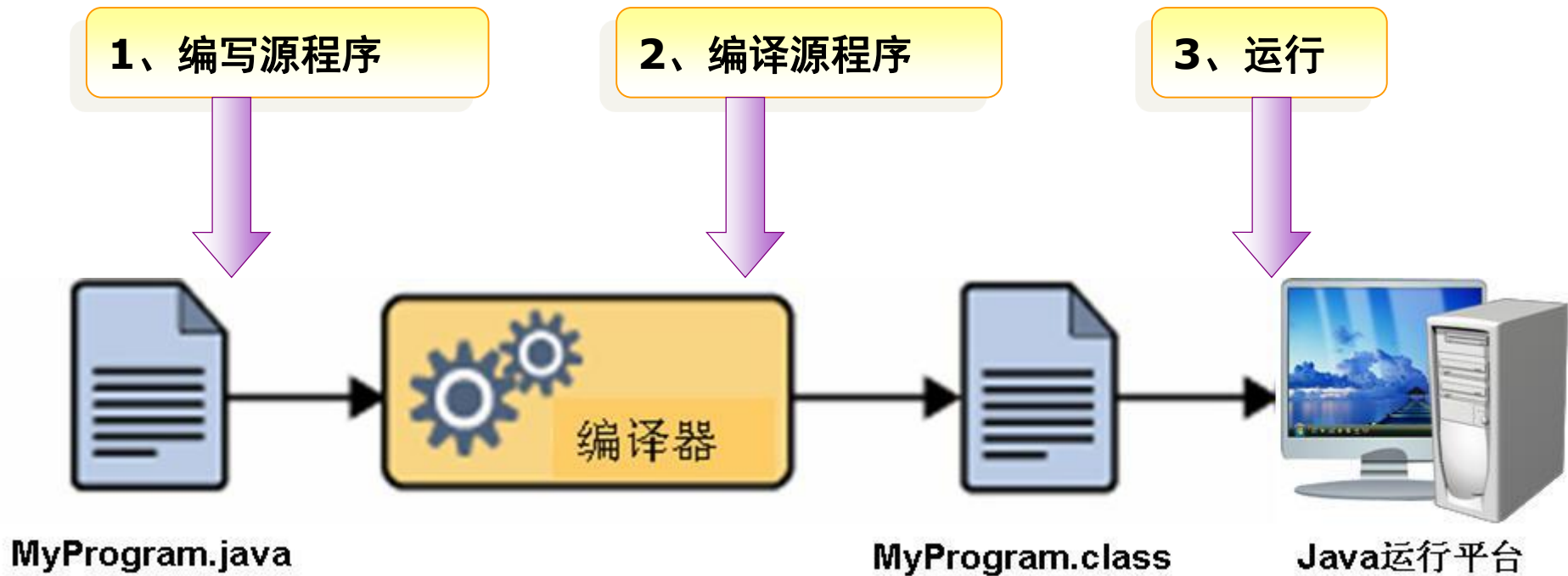
Java语言简介

Java开发环境构建

第一个Java程序

开发Java程序

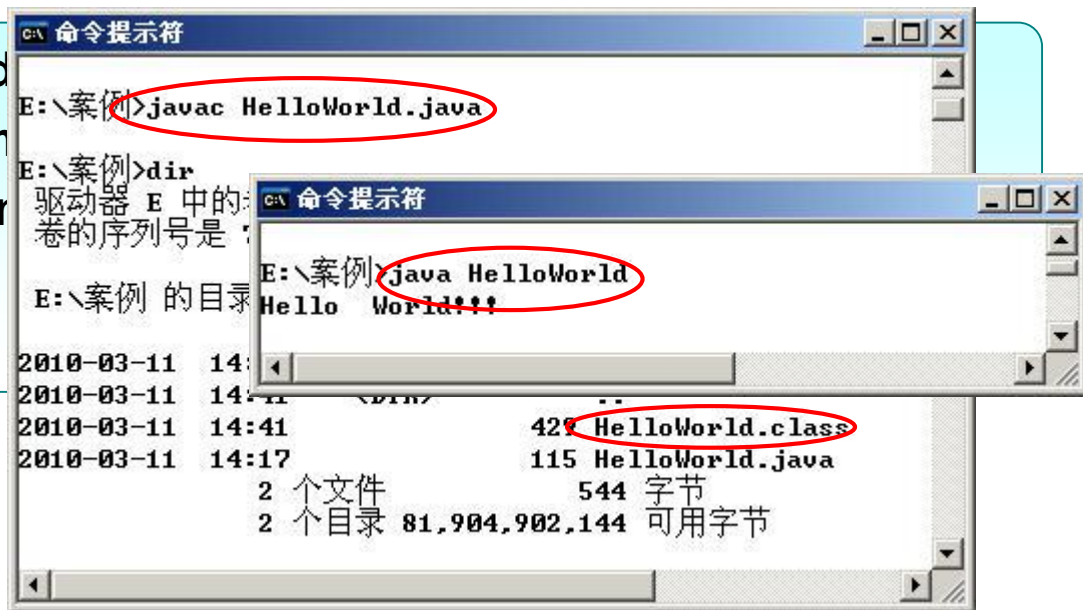
- 三步走



使用记事本开发Java程序的步骤

- 使用记事本编辑源程序，以.java为后缀名保存
- 使用javac命令编译.java文件，生成.class文件
- 使用java命令运行.class文件，输出程序结果

```
public class HelloWorld  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello World!!!");  
    }  
}
```



Java程序的结构



语法

外层框架

Java入口程序
框架

```
public class HelloWorld {
```

```
    public static void main(String[] args) {  
        System.out.println("Hello World!!!");  
    }
```

编写代码

```
}
```

Java程序的结构



分析

关键字

类名与文件名完全一样

main方法四要素必不可少

main方法是Java程序执行的入口点

```
public class HelloWorld{
```

```
    public static void main(String[] args){
```

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

```
    }
```

```
}
```

{和}一一对应，缺一不可

从控制台输出信息

从控制台输出信息2-1

- `System.out.println ()`

```
public class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.println("Hello World!!!");  
    }  
}
```

打印完引号中的信息后会自动换行

- `System.out.print()`

```
public class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.print("Hello World!!!");  
    }  
}
```

打印输出信息后不会自动换行



`System.out.println()`和`System.out.print()`有什么区别呢？

从控制台输出信息2-2



- 如何使System.out.println("");和System.out.print("\n"); 达到同样的效果?



○ 使用转义符

转义符	说 明
\n	将光标移动到下一行的第一格
\t	将光标移到下一个水平制表位置



```
public class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.print("Hello World!!!\n");  
    }  
}
```

打印输出信息后
将会自动换行

创建Java程序

- Java的源文件必须以扩展名.java作结束
- 主档名与类别名称必须一致
- 注意每个字母的大小写
- 空白只能是半型空格符或是Tab字符

编译Java程序

- `javac HelloJava.java`
- `error: cannot read: HelloJava.java`
 - `javac` 工具程序找不到您指定的.java 档案
- `HelloJava.java:1: class HelloJava is public, should be declared in a file named HellJava.java`
 - 类别名称与主档名不符

编译Java程序

- HelloJava.java:3: cannot find symbol
 - 程序代码中某些部份打错了，最常发生的原因可能是没有注意到字母大小写
- 'javac'不是内部或外部命令、可执行的程序或批处理文件
 - Path设定有误或没有在Path中加入JDK的「bin」目录

执行Java程序

- java HelloJava
- Exception in thread "main"
java.lang.NoClassDefFoundError
 - java工具程序找不到您所指定的类别
- Exception in thread "main"
java.lang.NoSuchMethodError: main
 - 没有指定Java程序的进入点（Entrypoint），java工具程序指定的类别必须要有一个程序进入点，也就是必须包括main(String[] args)这个方法（method）

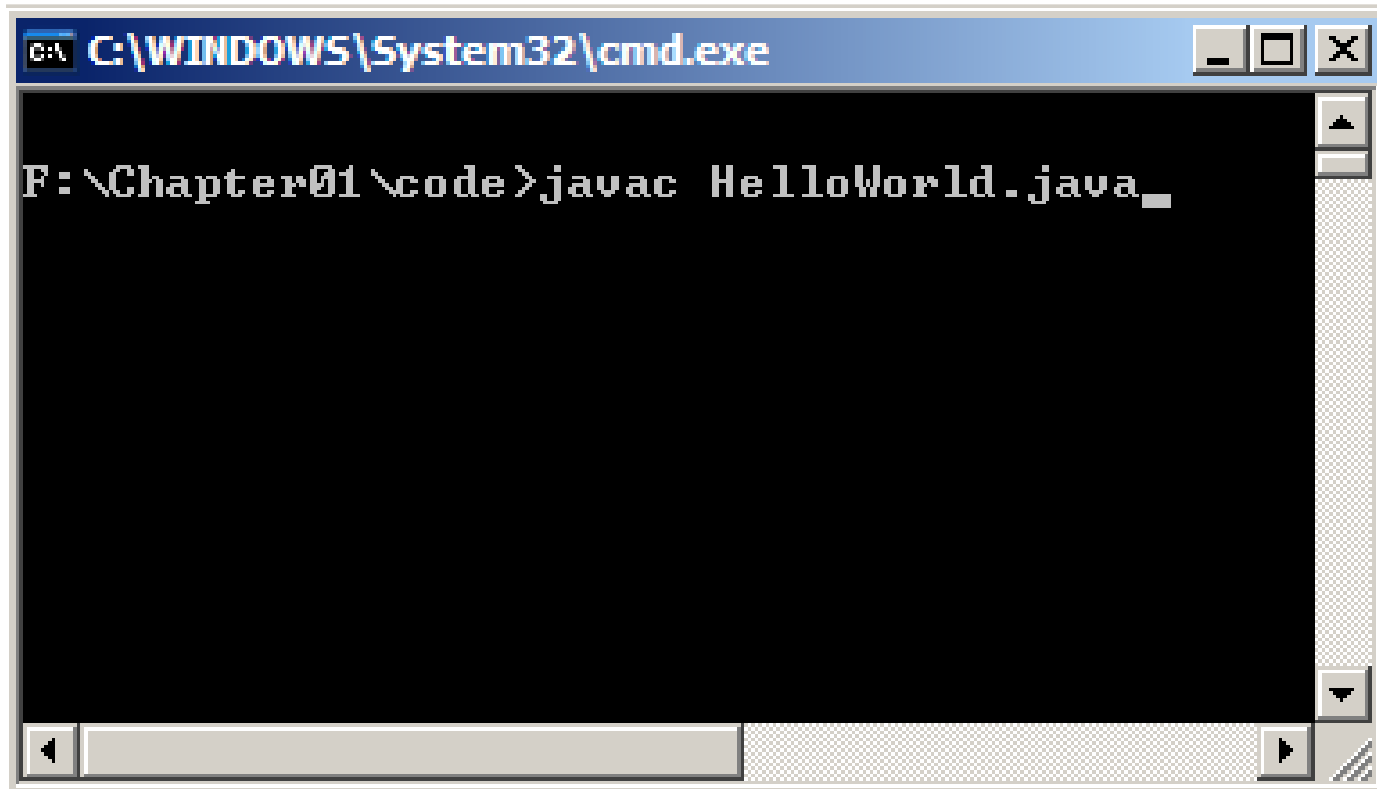
编写源代码注意事项

- ✓ 由于Java编译器和解释器对于代码中的字符是大小写敏感的，所以您在按照本书内容输入代码、命令以及文件名时，注意大小写是否正确。
- ✓ 该文件名必须是HelloWorld(和公开类的类名相同，并且大小写也要一样，这是由Java编译器和解释器是大小写敏感而决定的)，并且扩展名一定是.java。

编译源代码

- 在命令提示行输入以下命令并回车：

```
javac HelloWorld.java
```



观察

如果屏幕上没有出现错误提示，则表示已经正常完成了编译工作。再次使用DIR命令，可以发现目录下多了一个文件 HelloWorld.class，即编译好的中间字节代码(bytecode)文件



```
C:\WINDOWS\System32\cmd.exe

F:\Chapter01\code>dir
驱动器 F 中的卷没有标签。
卷的序列号是 909B-F097

F:\Chapter01\code 的目录

2004-12-28  16:05    <DIR>          .
2004-12-28  16:05    <DIR>          ..
2004-12-28  16:05                426 HelloWorld.class
2004-12-28  15:43                221 HelloWorld.java
               2 个文件                647 字节
               2 个目录 28,956,356,608 可用字节

F:\Chapter01\code>
```

常见问题解答

(1) Windows系统对于DOS命令中字符的大小写是不敏感的。例如：输入DIR、dir或是Dir得到的结果是相同的。

(2) Java编译器和解释器对于代码中的字符是大小写敏感的。所以在输入代码时要特别注意字符的大小写，例如HelloWorld和helloWorld是不同的，同样public和Public也是不同的。

常见问题一续1

(3) 保存源代码为文件时，要注意文件的命名。文件名一定要和源代码中公开类的类名一致(包括大小写)，并且文件扩展名必须为.java。注意，如果将例1.3.1中的public class HelloWorld改为class HelloWorld(即HelloWorld不再是公开类)，则可以将其保存为任何一个合法的、并以.java为扩展名的文件，例如NotHelloWorld.java。

常见问题一续2

(4) 如果在进行编译时报类似如下错误(如果读者使用的是英文版的操作系统, 错误信息将是英文):

'javac' 不是内部或外部命令, 也不是可运行的程序或批处理文件。

则表示系统不能正确定位编译器javac.exe, 原因是未能正确设定PATH环境变量。

常见问题一续3

编译时，需要用文件的全名，例如 `javac HelloWorld.java`；解释运行时只需要用类名，而不需要带上 `.class`，例如 `java HelloWorld`。如果输入：

```
java HelloWorld.class
```

那么系统报如下错误：

```
Exception           in           thread           "main"  
java.lang.NoClassDefFoundError:  
HelloWorld/class
```

理解例子

Java代码中具有三种类型的注释：

(1) 单行注释

```
// comments
```

从//至该行结束的内容是注释部分，编译器予以忽略。

(2) 多行注释

```
/* comments */
```

在/*和*/之间的所有内容均为注释部分，位于/*和*/之间的内容可以是一行或是多行。

(3) 文档注释

```
/ **   comments line 1
 *   comments line 2
 *   ...
 *   comments line n
 */
```

命令行参数&输出语句

- `public static void main(String[] args)`
- `System.out.println(String s)`

第一个Java程序

下面先看一个Java小程序，对Java编程有一个初步的认识。

【例1.1】求圆面积。

用文本编辑器(如Windows的记事本)编辑下列程序，文件名为Area.java。

Area.java

```
/** 求圆的面积 */
```

```
public class Area {
```

```
    double pi = 3.1415;
```

```
// 定义变量pi
```

```
    public static void main(String[] args) {
```

```
        double r,area;
```

```
        r = 3;
```

```
        area = pi * r * r;
```

```
// 求圆的面积
```

```
        System.out.println("圆的面积="+area);
```

```
// 在屏幕上显示结果
```

```
    }
```

```
}
```

第一个Java程序

下面对程序进行简要说明：

(1) `/**到*/`之间的内容为注释。

(2) 保留字`class`声明一个类，其类名为`Area`，保留字`public`表示它是一个公共类。类定义由花括号`{}`括起来。当编写一个Java源代码文件时，此文件通常被称为编译单元。每个编译单元都必须有一个后缀名为`.java`，而在编译单元内则可以有一个`public`类，该类的名称必须与文件的名称相同(包括大小写)。每个编译单元最多只能有一个`public`类，也可以没有，否则编译器就不会接受。

(3) 语句“`double pi = 3.1415;`”表示定义实型变量`pi`的值为3.1415。

(4) 在该类中定义了一个`main()`方法，其中`public`表示访问权限，指明所有的类都可以使用这一方法；`static`指明该方法是一个类方法，它可以通过类名直接调用；`void`则指明`main()`方法不返回任何值。对于一个应用程序来说，`main()`方法是必需的，而且必须按照如上的格式来定义。Java解释器在没有生成任何实例的情况下，以`main()`方法作为入口来执行程序。Java程序中可以定义多个类，每个类中可以定义多个方法，但是最多只能有一个公共类，`main()`方法也只能有一个。

Java Applet

Applet称为小应用程序，和应用程序不同的是，Applet一般是在支持Java的Web浏览器中运行。使用特定的HTML标签(tag)将Applet嵌入在HTML页面中，并且Applet程序和页面均存放在服务器上，当远程用户通过浏览器来访问页面时，页面中所嵌入的Applet程序通过网络被下载到浏览器中并加以运行

编写Applet源代码

```
import javax.swing.JApplet; //引入JApplet类
import javax.swing.JLabel; //引入标签类JLabel
public class HelloWorldApplet extends JApplet{
    public void init(){
        System.out.println("init Applet");
    }
    public void start(){
        System.out.println("start Applet");
        getContentPane().add(new JLabel("HelloWorldApplet"));
    }
    public void stop(){
        System.out.println("stop Applet");
    }
    public void destroy(){
        System.out.println("destory Applet");
    }
}
```


四个方法

- `init()` — 打开嵌有Applet的HTML页面时，该方法被调用以进行必要的初始化工作。
- `start()` — 该方法在`init`方法结束后被调用，用以启动Applet。
- `stop()` — 关闭嵌有Applet的HTML页面时，该方法自动调用，以关闭Applet。
- `destroy()` — `stop`方法执行完毕后，该方法被调用，用以释放Applet所使用的资源。

编写嵌入Applet的HTML文件

HelloToJava.html

```
<html>
<title>A Simple Applet Example</title>
<body>
This is a simple applet.<p>
<applet          code="HelloWorldApplet.class"
  width="150" height="150">
</applet>
</body>
</html>
```

运行Applet

- 运行应用程序使用的是java解释器。而运行小应用程序则使用Applet查看器或是WEB浏览器。
- (1) 使用appletviewer

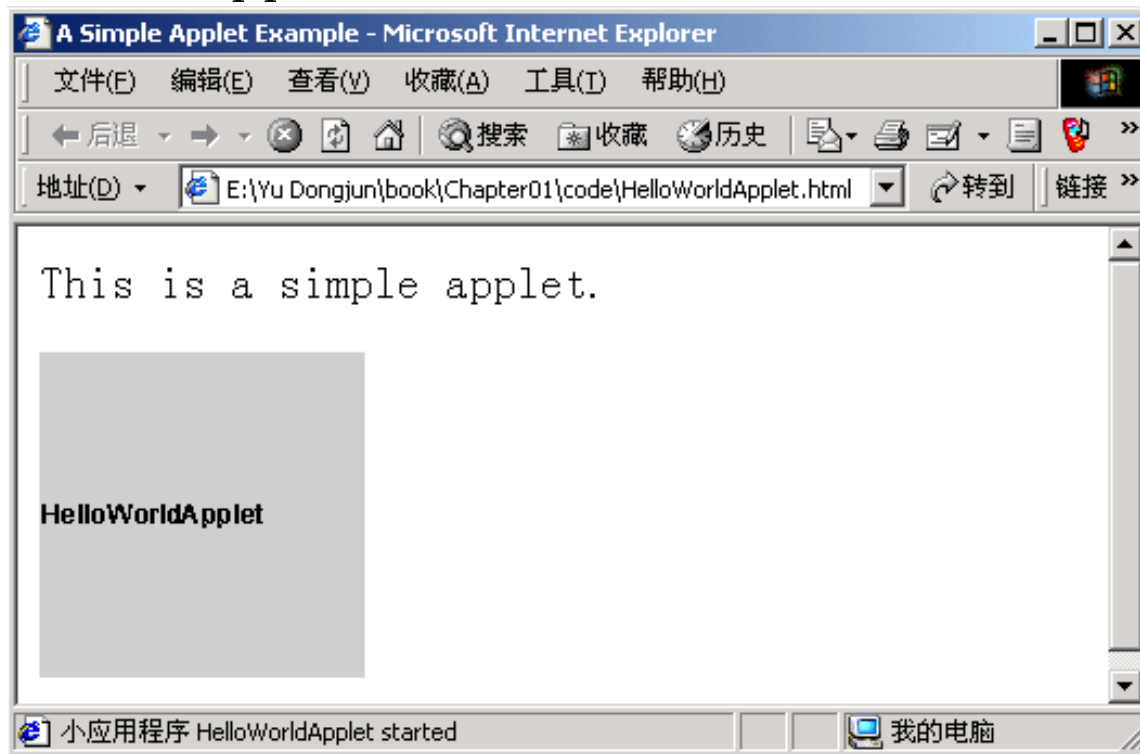
在DOS命令行输入如下命令并回车：

```
appletviewer HelloWorldApplet.html
```



运行Applet-续1

- 使用WEB浏览器，例如Internet Explorer
直接使用WEB浏览器打开页面文件
HelloWorldApplet.html



选择开发工具

- 从简单的文字编辑辅助工具开始
 - UltraEdit (<http://www.ultraedit.com/>)
 - Editplus (<http://www.editplus.com/>)
- 简单的开发环境
 - JCreator (<http://www.jcreator.com/>)
 - BlueJ (<http://www.bluej.org/index.html>)
- 功能更齐全的IDE
 - Eclipse (<http://www.eclipse.org/>)
 - NetBeans (<http://www.netbeans.org/>)