JAVA程序设计

贡正仙
zhxgong@suda.edu.cn

主要内容

- Java运行环境
- Java程序类型

Java运行环境

- Java语言的语法和构成是稳定的
- Java API一直在扩展
- Sun公司使用JDK这一Java开发工具箱
  - JDK是一个简单的命令行工具集
    - 包括软件库、编译Java源代码的编译器、执行Java字节码的解释器、测试Java Applet的浏览器，以及其他实用工具
  - JDK包含Java运行环境

Java的运行环境

- 最基本的模式：
  - JDK(JSDK)包＋文本编辑器
  - 软件安装：
    - JDK——选择恰当的操作系统（不同操作系统上有不同的JDK，但安装完毕后，不论什么平台上编译得到的Java字节码，不需重新编译即可运行）
    - 配置执行路径：Win2k及以上版本的系统中，在（控制面板→系统→环境变量→用户变量中的PATH）里添加JDK安装目录中的bin子目录的路径

Java SE Downloads

Java Platform (JDK) 8u31

NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u31

This release includes important security fixes. O
users upgrade to this release.
Learn more

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
  - JDK ReadMe
  - JRE ReadMe

Java SE Development Kit 8u31

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ **Accept License Agreement** ◉ **Decline License Agreement**

Product / File Description	File Size	Download
Linux x86	135.24 MB	jdk-8u31-linux-i586.rpm
Linux x86	154.91 MB	jdk-8u31-linux-i586.tar.gz
Linux x64	135.62 MB	jdk-8u31-linux-x64.rpm
Linux x64	153.45 MB	jdk-8u31-linux-x64.tar.gz
Mac OS X x64	209.17 MB	jdk-8u31-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	136.91 MB	jdk-8u31-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.11 MB	jdk-8u31-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.51 MB	jdk-8u31-solaris-x64.tar.Z
Solaris x64	94.82 MB	jdk-8u31-solaris-x64.tar.gz
Windows x86	157.96 MB	jdk-8u31-windows-i586.exe
Windows x64	170.36 MB	jdk-8u31-windows-x64.exe

Java的运行环境

- 最基本的模式：
  - 软件安装：
    - Java类库路径的设置（CLASSPATH）：
      - 安装目录的lib子目录中的tools.jar
      - 说明：JDK1.3以后的版本能够自动定位Java类库的位置，因此不再需要设置CLASSPATH
    - 浏览Java安装完毕后的目录：

JDK目录树

Bin	编译器、解释器以及一些工具
Docs	库文档，HTML格式
Demo	演示程序
Include	用于本地方法的文件
Lib	库文件
Jre	Java运行环境文件（JVM和运行类库等）
src	库源文件的各个子目录，src.zip文件

JDK中包含的基本开发工具

javac.exe	Java编译器，将源代码编译成字节码
java.exe	解析器，执行字节码
Javap.exe	反编译，将Java类文件还原成方法和变量
appletviewer.exe	测试运行Applet小应用程序
jar.exe	压缩打包，负责将类文件和其他资源绑定成jar文件
javadoc.exe	文档生成器，由Java源文件生成相应的HTML页面，对源文件中类进行内部索引

回顾

Java程序的执行流程:

编译阶段：
Java源文件（*.java）->Java编译器->字节码文件（*.class）

运行阶段：
字节码文件（*.class）->类装载器-》字节码校验器-》解释器-》
操作系统将整个文件load到内存区，找到main方法开始实行

- **1.4　Java程序的开发步骤**

1．编写源文件:扩展名必须是.java。

2．编译Java源程序:用Java编译器（javac.exe）编译源文件，得到字节码文件。

3．运行Java程序:使用Java解释器（java.exe）来解释执行字节码文件。

• 1.5 简单的Java应用程序

• 1.5.1 源文件的编写与保存

Java应用程序的源文件是由若干个书写形式互相独立的类组成。

例子1中的Java源文件Hello.java是由两个名字分别为Welcome和Student的类组成。

```
例子1
public class Welcome{
   public static void main (String args[]){
      System.out.println(“this is a simple java application");
      Student stu=new Student();
      stu.speak(“Welcome,new students!");
   }
}
class Student{
   public void speak(String s){
      System.out.println(s);
   }
}
```

1.5 简单的Java应用程序

- Java应用程序：
  - Java源程序文件名：
    - 可以包含多个class类，若里面无public类，则符合系统命名的文件名都可以为源程序文件名
    - 若包含public类，则必须与public类同名，**并且大小写必须相同**
    - 源程序后缀为java，即前例源程序名必须为：Welcome.java

• 1.6 Java应用程序的基本结构

一个Java应用程序（也称为一个工程）是由若干个类所构成，这些类可以在一个源文件中，也可以分布在若干个源文件中，如图1.12所示。

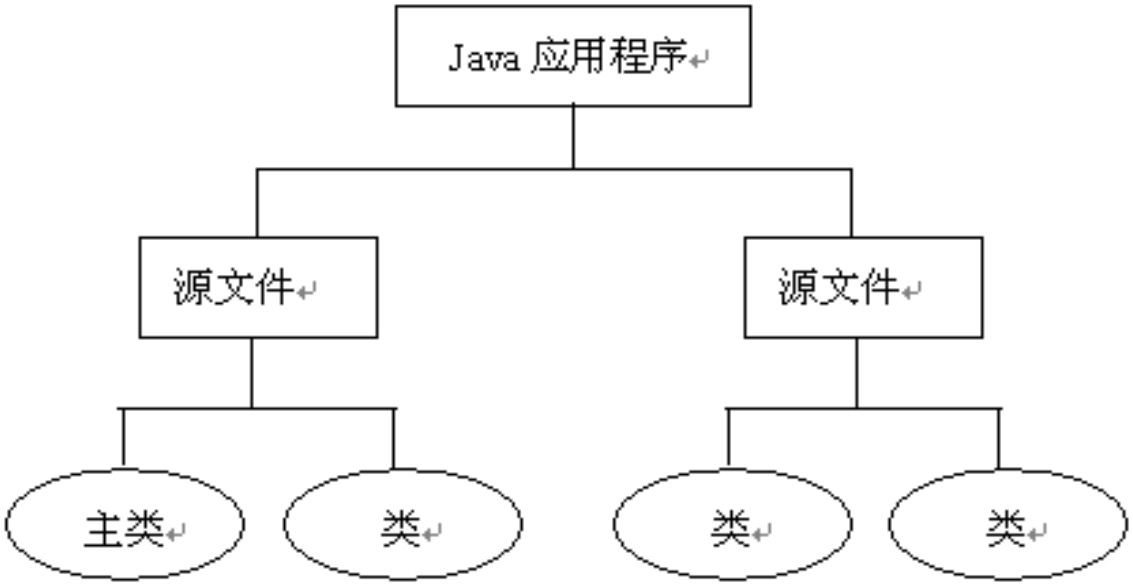


图 1.12 程序的结构

1.7 Java编译与运行

- 编译Java源文件：
  - 指令格式：javac [options] [sourcefiles]
  - 主要选项：
    - -nowarn：不输出警告
    - -classpath <路径>：指定用户设定的classpath
    - -sourcepath <路径>：指定源文件的路径
    - -d <目录>：指定编译生成的class文件存放的目录
  - 例如：

```
javac –sourcepath c:\test\src –classpath
   c:\test\classes –d c:\test\classes
   c:\test\src\com\*.java   c:\test\src\coll\Doll.java
```

//编译c:\test\src\com下的所有java文件，和coll下的Doll.java文件

1.7 Java编译与运行

- 运行Java程序：
  - 格式：java <options> <类文件主名>
  - 主要选项：
    - -classpath <路径>：覆盖classpath变量
    - -jar：指定运行某个jar文件中的特定java类

1.7 Java编译与运行

- Java应用程序：
  - 示例说明：
    - 执行编译得到的字节码文件
      - 命令：java <字节码文件主名>
      - 例如前例编译后的执行指令为：

        java　Welcome

        千万注意不要 **java Welcome.class!**

编程练习

- 创建一个Java源程序，按下列步骤编译并运行：
  - 创建名为Welcome.java文件，可使用任意编译器，将其保存成文本格式
  - 编译源程序
  - 运行字节码
  - 将main替换成Main，进行测试
  - 使用javac welcome.java，查看结果
  - 使用java welcome，查看结果
  - 使用java Welcome.class，查看结果

1.8Java的集成开发环境 （eclipse）

- 集成开发环境
  - 新建工程
  - 添加类
  - 编辑类
  - 设定主函数（Applet除外）
  - 编译运行

1.3 Java程序运行环境配置与使用

1.3.4 使用集成开发环境

1. Eclipse概述

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。它专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台，主要由Eclipse项目、Eclipse工具项目和Eclipse技术项目3个项目组成。

1.3 Java程序运行环境配置与使用

1.8 使用集成开发环境

1. Eclipse获取与安装

Eclipse是一个开放源代码的项目，可以到其官方网站www.eclipse.org上免费下载Eclipse的最新版本。

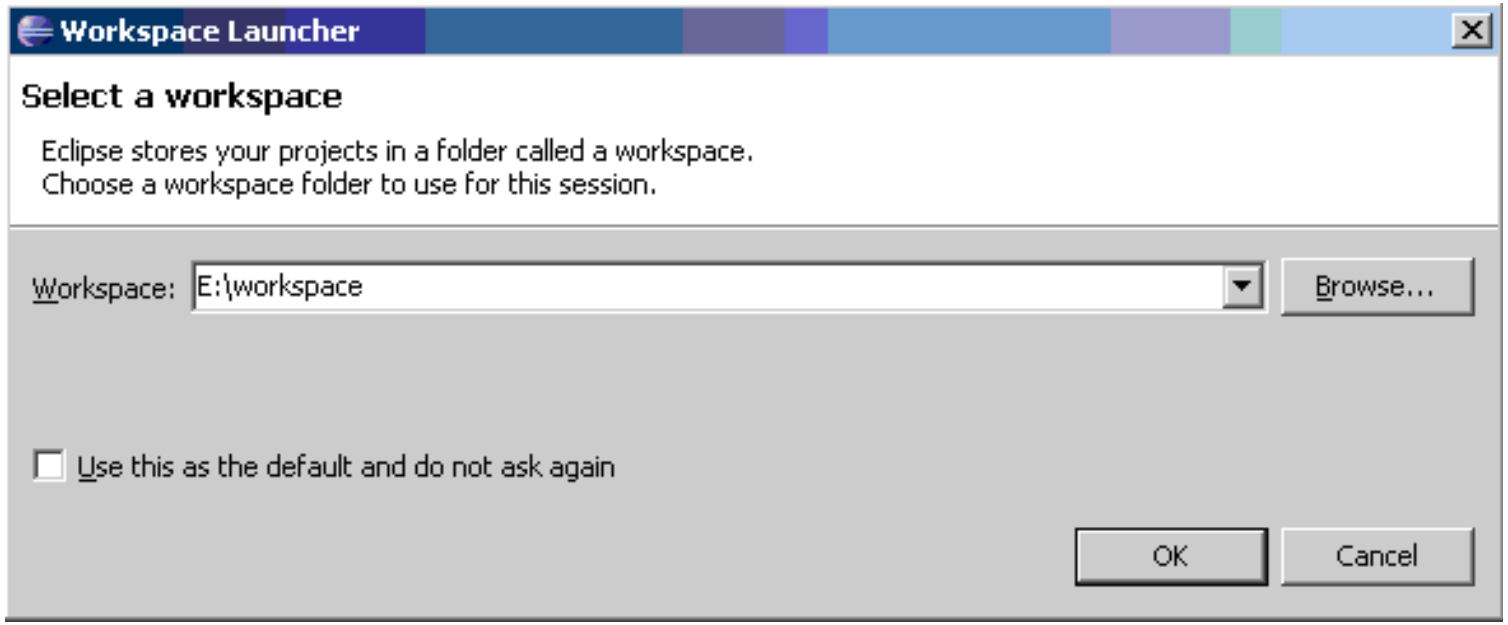
在解压后，可以到相应的安装路径下找到Eclipse.exe文件双击运行。

1.3 Java程序运行环境配置与使用

1.8 使用集成开发环境

2. Eclipse获取与安装

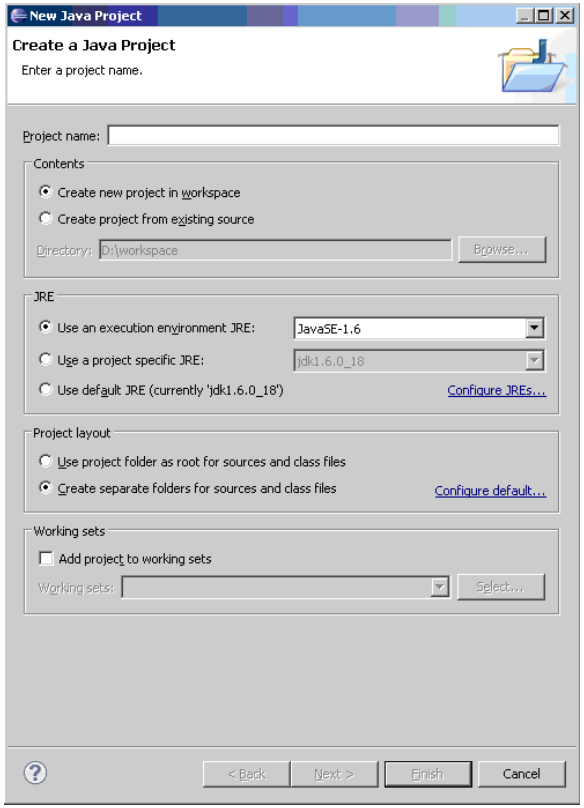
随后出现一个如图所示的选择工作区路径对话框。Eclipse会将所有文件存放在工作区指定的路径下：

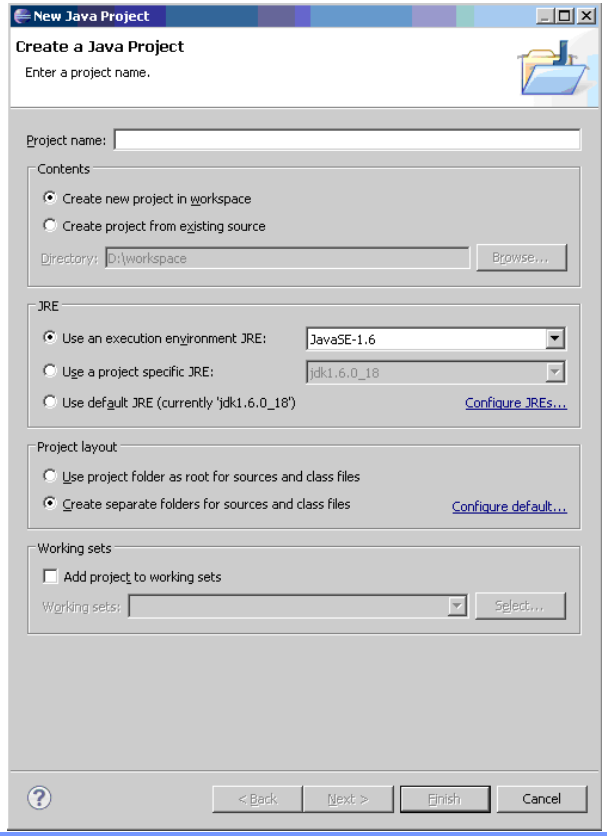


1.3 Java程序运行环境配置与使用

1.8 使用集成开发环境

3. Eclipse开发Java程序



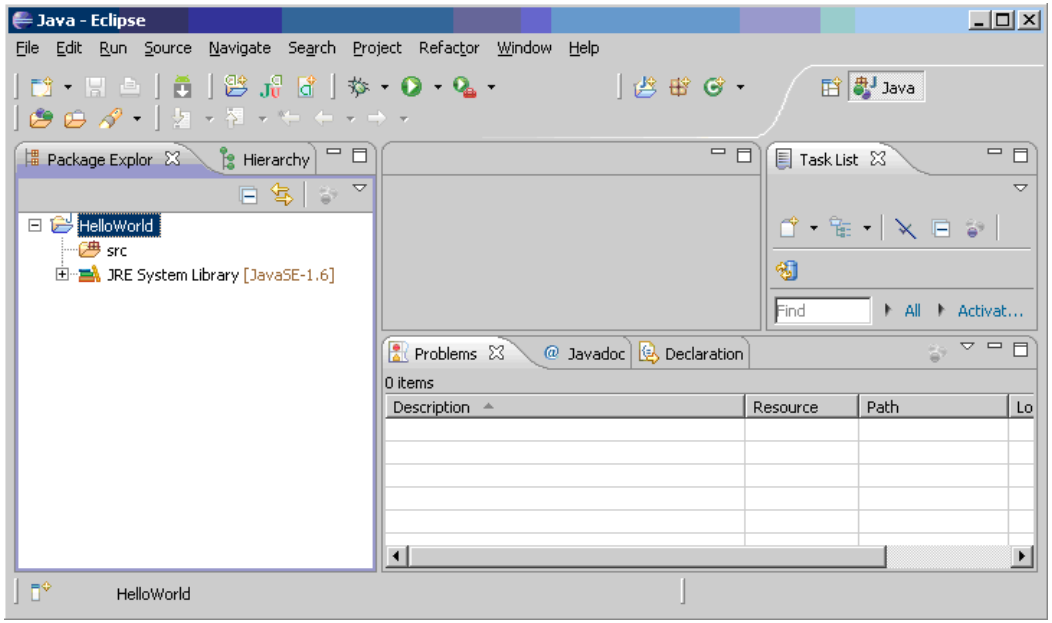


1.3 Java程序运行环境配置与使用

1.8使用集成开发环境

3. Eclipse开发Java程序

在该窗口中，可以对Java项目进行设置。如设置该项目需要添加的项目，需要而外引入的类库文件等。直接点击【Finish】按钮，Eclipse就会自动创建一个HelloWorld项目，如图所示。

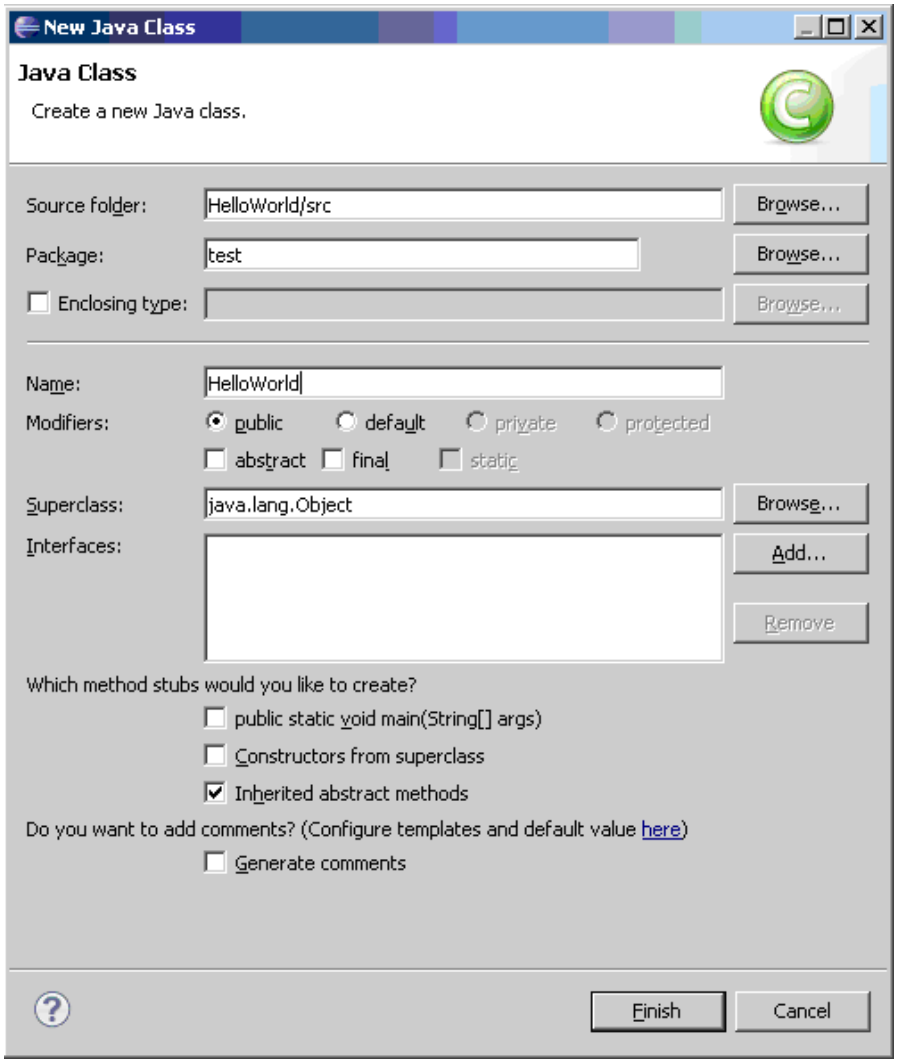


1.3 Java程序运行环境配置与使用

1.8 使用集成开发环境

3. Eclipse开发Java程序

项目创建完成后，就可以直接在该项目中创建文件，执行【File】|【New】|【Class】命令，或者选中HelloWorld右击，在弹出菜单选择【New】|【Class】选项，弹出如图所示的窗口。

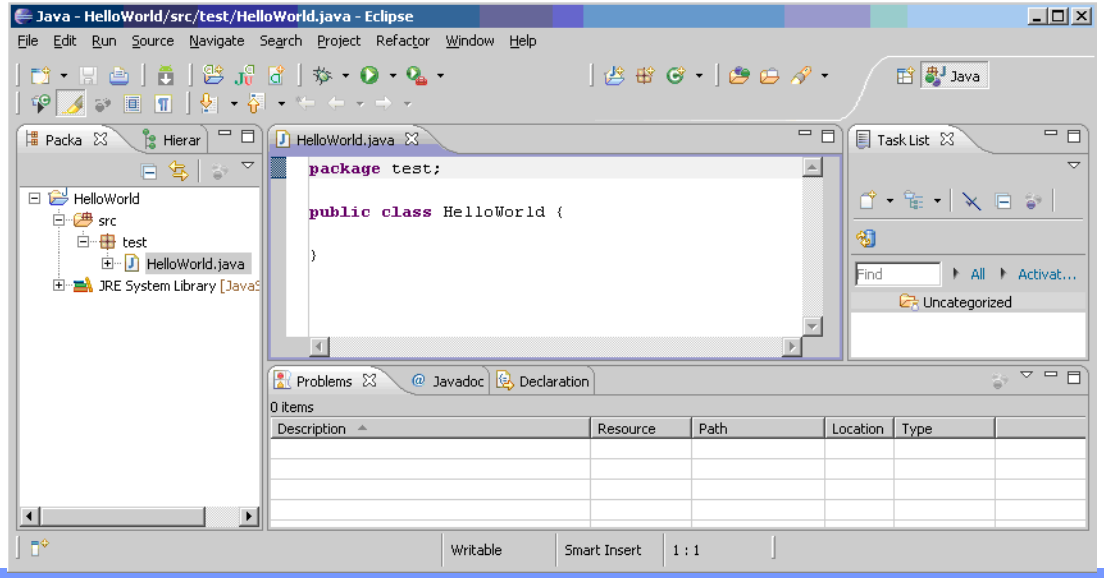


1.3 Java程序运行环境配置与使用

1.8使用集成开发环境

3. Eclipse开发Java程序

在该窗口中的【package】文本域中，输入该类所在包的名称。该名称一般为小写字母，如果输入的是大写字母，或没有输入，就会弹出提示消息。在【name】文本域中输入要创建的类的名称。该名称第一个字母需大写。配置包名与类名后，单击【完成】按钮，出现如图所示窗口。

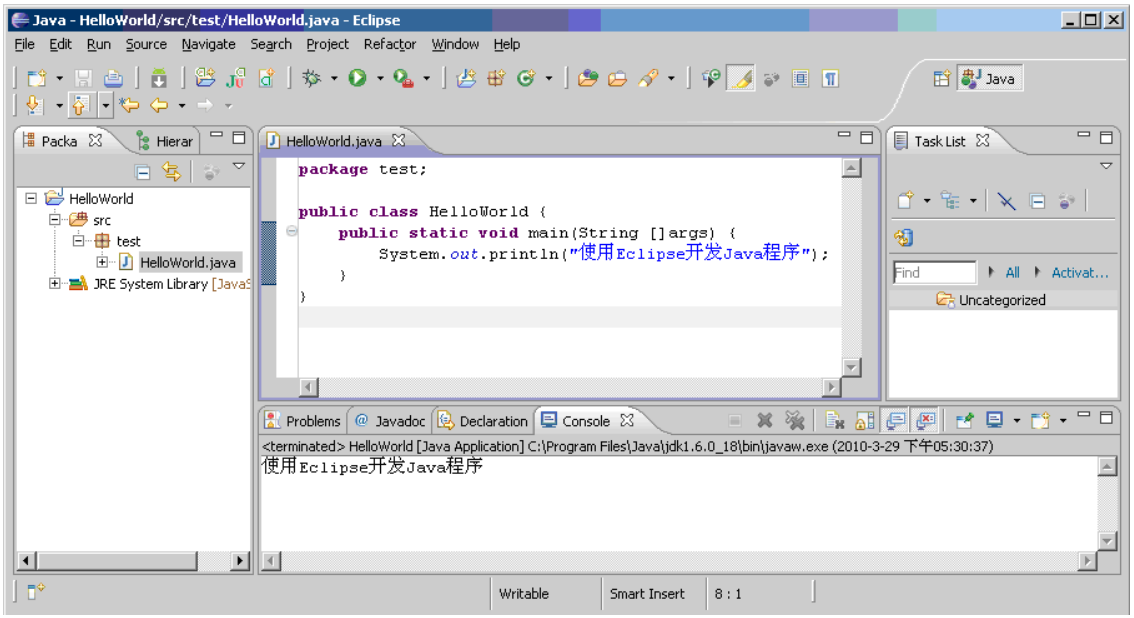


1.3 Java程序运行环境配置与使用

1.8使用集成开发环境

3. Eclipse开发Java程序

现在，可以在Eclipse平台上开始编辑Java程序。在输入时注意到，Eclipse编辑器提供一些特性，包括语法检查和代码自动提示功能。如图所示。

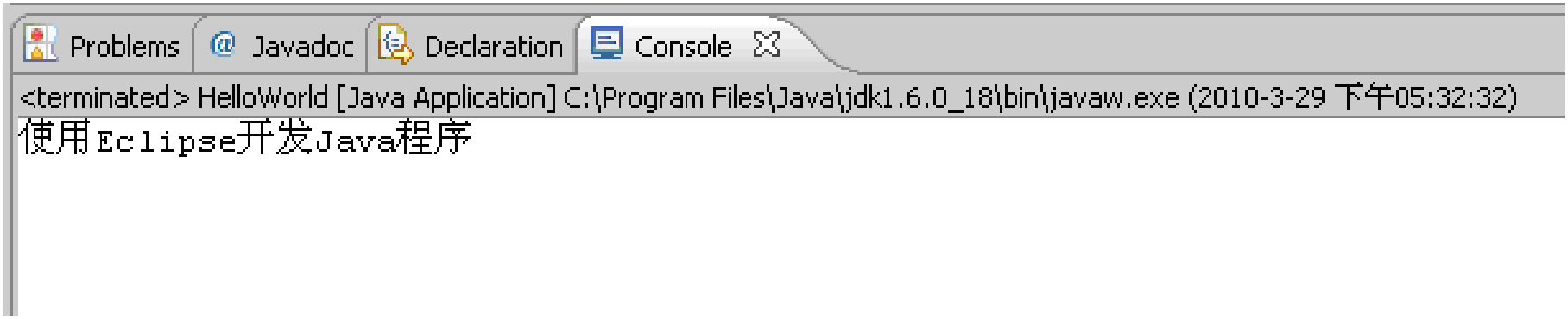


1.3 Java程序运行环境配置与使用

1.8 使用集成开发环境

3. Eclipse开发Java程序

编辑完成后，运行该程序。单击Eclipse工具栏上的【Console】 图标，会在Eclipse控制台显示程序结果，如图所示。



补充

Java程序的类型

- 两类基本的Java程序：
  - Java应用程序
    - 可独立运行，公共类public中包含main主函数的一类Java程序
  - Java Applet小应用程序
    - 无main主函数，必须嵌在某个HTML文件中，通过浏览器运行的一类Java程序

Java程序初步

- Java小应用程序——Java Applet
  - 几点说明：
    - 小应用程序不能独立运行，必须嵌入到HTML页面
    - 运行过程：
      - 先利用javac编译程序将源程序编译成*.class文件（源程序命名规则与普通Java应用程序一样，与public类同名）
      - 再编写对应的HTML页面
      - 然后用appletviewer浏览applet的运行，运行指令为：appletviewer <HTML文件名>

一个简单的Applet程序


```
import java.applet.Applet;
import java.awt.*;
public class MyApplet extends Applet {
public void paint(Graphics g)
    { g.setColor(Color.red);
      g.drawString("java applet 1",5,30);
      g.setColor(Color.blue);
      g.drawString("java applet 2",10,50);

    }
}
```

一个简单的Applet程序的调用

Applettest.html的内容：

```
<html>
<title> An Example Homepage </title>
<body>
<h1> Welcome to my homepage! </h1>
you can see an applet in it。<p>
<br>
<applet code="MyApplet.class" width = 300 height=300>
</applet>
</body>
</html>
```

```
C:\javaclass\helloworld\bin>appletviewer applettest.html

C:\javaclass\helloworld\bin>
```

编程练习

- 创建一个HTML文件，调用Applet在Web浏览器中显示Welcome to java!
  - 创建名为WelcomeApplet.java的Applet
  - 创建名为WelcomeApplet.html的HTML文件
  - 编译WelcomeApplet.java文件
  - 在浏览器中查看WelcomeApplet.html
  - 利用appletviewer查看applet

几个注意点

Java源程序：

- 可以没有public类
- 若有public类，只能有一个，public类名与java主文件名相同

Javac 的用法补充

javac A.java B.java

javac *.java

Javac –classpath .;c:\xx;d:\yy *.java

Java A 在当前路径找A类

注意 classpath

- 不设置classpath

  系统默认寻找次序： 当前，环境变量

- 设置classpath，按照classpath,注意 . 的设置

  java –classpath .;c:\xx;d:\yy A