

## 第2章 简单的JAVA程序



- 2.1 JAVA程序的类型
- 2.2 开发JAVA 程序的基本步骤
- 2.3 基本输入输出编程
- 2.4 JAVA编程工具

# Java程序的类型





# Java程序的类型



# Application和Applet程序



- 结构和运行环境不同
- 前者是独立的程序，需要执行器(调用虚拟机)来运行
- 后者是嵌在HTML网页中的非独立的程序，
  - 由专门的appletViewer来运行
  - 或者由Web 浏览器（调用JAVA虚拟机）来运行

# Application程序



- HelloWorld.java

- 要点

- class 是主体
- public类名与文件同名
- main()的写法是固定的
- System.out.print 及 println 及 printf

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

# Applet程序



- HelloWorldApplet.java

- import表示导入
- extends JApplet表示继承
  - Applet或JApplet都可以
- 有paint()方法，表示如何绘制
- 没有main()方法

- HelloWorldApplet.html

```
import java.awt.*;
import java.applet.*;
import javax.swing.*;
public class HelloWorldApplet extends JApplet {
    public void paint(Graphics g){
        g.drawString ("Hello World!",20,20);
    }
}
```

```
<HTML>
<HEAD><TITLE> An Applet </TITLE></HEAD>
<BODY>
<applet code="HelloWorldApplet.class"
        width=200 height=40 background=white>
</applet>
</BODY>
</HTML>
```

# JAVA程序的基本构成



- HelloDate.java

- package 语句 ( 0或1句 )
- import 语句 ( 0或多句 )
  - 导入其他类的类库
- 类定义——class ( 1或多个 )
  - 一个文件只能有一个public类(与文件同名)

```
package edu.pku.tds.ch02;  
  
import java.util.*;  
  
public class HelloDate {
```

- 类 = 类头 + 类体
- 类成员 = 字段 ( field ) + 方法 ( method )
  - 字段 ( field, 属性, 变量 )      方法 ( method, 函数 )
- 方法 = 方法头 + 方法体



# Java程序的编译与运行





# 程序的编辑、编译与运行



- 源程序编辑
  - 可用任一文本编辑器
- 程序编译
  - 使用JDK中的`javac`工具
- 程序运行
  - 使用`java`工具

# JAVA工具包JDK



- Java 编程的基本工具是JDK
- 下载 <http://java.sun.com>
  - ▣可单独下载JavaSE，也可以下载与Netbeans绑定的版本
- JDK安装后的文件夹
  - ▣Bin 该目录存放工具文件
  - ▣Jre 该目录存放与java 运行环境相关的文件
    - 注：该 Jre与从<http://java.com> 下载的JRE略有区别
  - ▣Demo 该目录存放一些示例文件
  - ▣Include 该目录存放与C相关的头文件
  - ▣Lib 该目录存放程序库
  - ▣Db 数据库相关



# Application的编辑、编译与运行

- 程序编辑：编辑器——文件名要与public class的类名一致
  - 区分大小写
- 程序编译——转换为字节码 ( bytecode)文件，扩展名.class
  - ( .class文件中包含 java虚拟机的指令 )
  - 编译可以使用JDK工具javac.exe。
  - 如 **javac** Hello.java
- 程序的运行——执行 .class文件中的指令的过程。
  - 如 **java** Hello
  - (注意：不要写成 **java Hello.class**，因为这里需要的是类名，不是文件名)



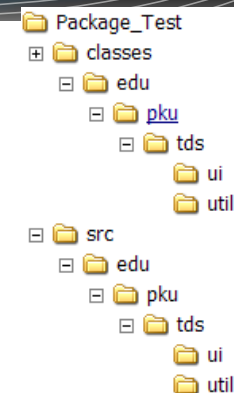
# 设定path和classpath

- 设定path和classpath
  - 前者是命令 ( javac及java)的路径; 后者是所要引用的类的路径
  - 可以在命令行上设定
    - set path=.;c:\jdk\bin;...
  - 也可以在系统环境中设定
    - 如win7中：我的电脑—属性—高级—性能—环境变量
    - 如win8中：这台电脑—属性—高级—环境变量
- 在javac及java命令行上使用-classpath ( 或-cp)选项可以引用别的库
  - javac -cp libxx.jar 源文件名.java
  - java -cp libxx.jar 类名



# 使用package时的编译

- 文件及路径一致
- 程序中使用package语句
- 使用import语句



- 编译及运行

□ `javac -d classes src\edu\pku\tds\ui\*.java src\edu\pku\tds\util\*.java src\edu\pku\tds\*.java`

□ `java -cp classes edu.pku.tds.PackageTest`



# Applet的编辑、编译与运行

- Java Applet程序必须嵌入到 HTML中，并由负责解释HTML 文件的 WWW 浏览器充当解释器，解释执行程序。
- Java Applet在WWW 中引入了动态交互的内容。
- 1、源程序的编辑和编译。
- 2、在HTML文件中嵌入Applet。
  - 使用<applet>标签:
  - `<applet code="HelloWorldApplet.class"`
  - `width=200 height=40 background=white>`

# 用appletViewer运行Applet



- appletViewer HelloWorldApplet.html







# 用浏览器浏览applet

- 首先从 <http://java.com> 下载java ( JRE ) 并安装
- 启用浏览器中Java
  - 控制面板——程序--java——安全——中，并启用浏览器中的Java
- 将.class及.html文件放到www服务器上，然后用浏览器访问
  - 从Java8开始，Applet的运行受到更严格的限制
  - 如不能在本机文件 ( file:// ) 中运行applet，必须在http://或https://
- 更多的Applet替代方案
  - Flash，SilverLight等



# 其他几个工具



- 主要的工具
  - javac 编译
  - java 运行（控制台及图形界面程序）
  - javaw 运行图形界面程序
  - appletViewer 运行applet程序
- 另外常用的几个工具
  - jar 打包工具
  - javadoc 生成文档
  - Javap 查看类信息及反汇编



# 使用jar打包

- ( 1 ) 编译 `javac A.java`
- ( 2 ) 打包 `jar cvfm A.jar A.man A.class`
  - ▣ c表示创建(create), v表示显示详情(verbose), f表示指定文件名, m表示清单文件
- ( 3 ) 运行 `java -jar A.jar`
- 其中A.man 是清单文件(manifest), 内容如下 :
  - Manifest-Version: 1.0
  - Class-Path: .
  - **Main-Class:** A
- 清单文件可以任意命名 , 常见的是用 MANIFEST.MF



# 使用JavaDoc生成文档

- javadoc -d 目录名 xxx.java
- /\*\* \*/ 这其中可以用以下标记
  - @author 对类的说明 标明开发该类模块的作者
  - @version 对类的说明 标明该类模块的版本
  - @see 对类、属性、方法的说明 参考转向，也就是相关主题
  - @param 对方法的说明 对方法中某参数的说明
  - @return 对方法的说明 对方法返回值的说明
  - @exception 对方法的说明 对方法可能抛出的异常进行说明

# Java的API文档



- 在线文档
  - <http://docs.oracle.com/javase/8/docs/api/index.html>
- 也可以下载网页格式的文档
  - doc.zip
- 另可以从网上搜索到chm格式的文档
  - 如 JDK\_API\_1\_6\_zh\_CN.CHM



# 使用javap

- 使用javap查看类的信息

- ▣ javap 类名

- 使用javap反汇编

- ▣ javap -c 类名

```
D:\JavaExample\ch02>javap A
Compiled from "Test_Virtual_Static.java"
class A {
    A();
    static void f(int);
}

D:\JavaExample\ch02>javap -c A
Compiled from "Test_Virtual_Static.java"
class A {
    A();
    Code:
        0: aload_0
        1: invokespecial #1
    ()V
        4: return

    static void f(int);
    Code:
        0: getstatic     #2
io/PrintStream;
        3: new           #3
        6: dup
```



# 程序的输入与输出



# 输入与输出



- 应用程序 ( Java Application ) 的输入输出可以是文本界面，也可以是图形界面。
- 小程序 ( Java Applet ) 则只能是图形界面。
- 每种界面都可以有输入和输出。





# 文本界面：使用Scanner类

- 使用java.util.Scanner类

- 用其nextInt()方法
- 还有nextDouble()
- next()得到下一个单词
- 注：Since JDK1.5

- ScannerTest.java

```
import java.util.Scanner;
class ScannerTest{
    public static void main( String[] args ){
        Scanner scanner = new Scanner(System.in);
        System.out.print("请输入一个数");
        int a = scanner.nextInt();
        System.out.printf("%d的平方是%d\n",a,a*a);
    }
}
```





# 文本界面：使用in及out

- java.io包
- System.in.read( )
- System.out.print( ) 及 println、 printf （类似于C语言
- AppCharInOut.java
  - 注意 try{}catch

```
char c = ' ';  
System.out.print("Please input a char: ");  
try{  
    c = (char) System.in.read();  
  
}catch(IOException e){}  
System.out.println("You have entered: " + c );
```

# 示例



- AppLineInOut.java

- 输入输出行
- 更复杂一些

- AppNumInOut.java

- 输入输出数字
- 用Integer.parseInt( s );
- 用Double.parseDouble(s)

```
try{  
    BufferedReader in = new BufferedReader(  
        new InputStreamReader( System.in ) );  
    s = in.readLine();  
}catch(IOException e){}
```

```
BufferedReader in = new BufferedReader(  
    new InputStreamReader( System.in ) );  
System.out.print("Please input an int: ");  
s = in.readLine();  
n = Integer.parseInt( s );
```



# 图形界面输入与输出

- 在图形界面下,可以:
  - 使用文本框对象 ( TextField) 获取用户输入的数据。
  - 使用标签对象(Label)或文本框对象输出数据。
  - 使用命令按钮(Button)来执行命令。



# JAVA Application 图形界面输入与输出

Java程序设计



- Java Application需要首先创建自己的图形界面.
- [AppGraphInOut.java](#)
- 通过创建一个Frame创建自己的用户界面,在构建AppFrame时,设定该Frame的大小,并用setVisible(true)方法显示出来.



- AppGraphInOut.java

- add(xxxx) 加入对象

- btn.addActionListener

- 处理事件

- actionPerformed ( ) 函数

- 具体处理事件

```
setLayout( new FlowLayout() );
add( in );
add( btn );
add( out );
btn.addActionListener( new BtnActionAdapter() );
}

class BtnActionAdapter implements ActionListener
{
    public void actionPerformed((ActionEvent e)
    {
        String s = in.getText();
        double d = Double.parseDouble( s );
        double sq = Math.sqrt(d);
        out.setText( d + "的平方根是: " + sq );
    }
}
```



- 在Java8中可以简写为  
    □ `e->{ ... }`
- [AppGraphInOut8.java](#)

```
btn.addActionListener( e->{  
    String s = in.getText();  
    double d = Double.parseDouble( s );  
    double sq = Math.sqrt(d);  
    out.setText( d + "的平方根是: " + sq );  
});
```

# Applet输入输出



- AppletInOut.java
- 在init()中
  - add(xxxx) 加入对象
  - btn.addActionListener
    - 处理事件
- actionPerformed ( ) 函数
  - 具体处理事件

```
setLayout( new FlowLayout() );
add( in );
add( btn );
add( out );
btn.addActionListener( new BtnActionAdapter() );
}

class BtnActionAdapter implements ActionListener
{
    public void actionPerformed( ActionEvent e )
    {
        String s = in.getText();
        double d = Double.parseDouble( s );
        double sq = Math.sqrt(d);
        out.setText( d + "的平方根是: " + sq );
    }
}
```



# 同时作为 Application和Applet 的程序



- 三个条件:
  - 是Applet的派生;
  - 含有main();
  - 在main()中创建一个用户界面,并将这个Applet加入.
- [AppAppletInOut.java](#)





# 集成开发工具



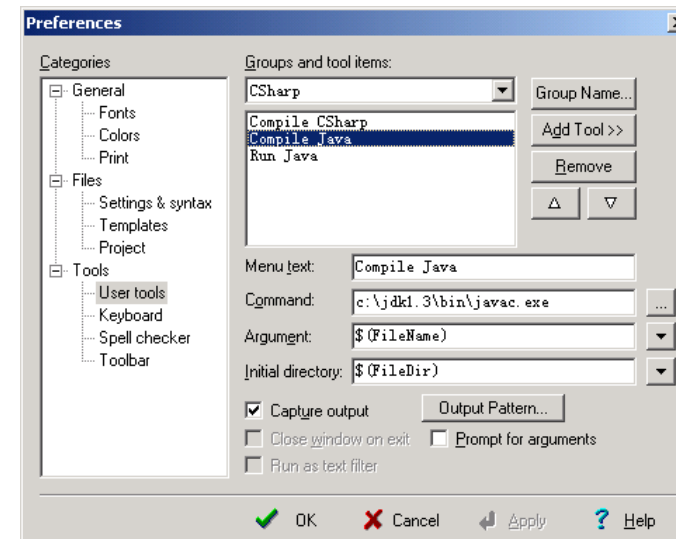
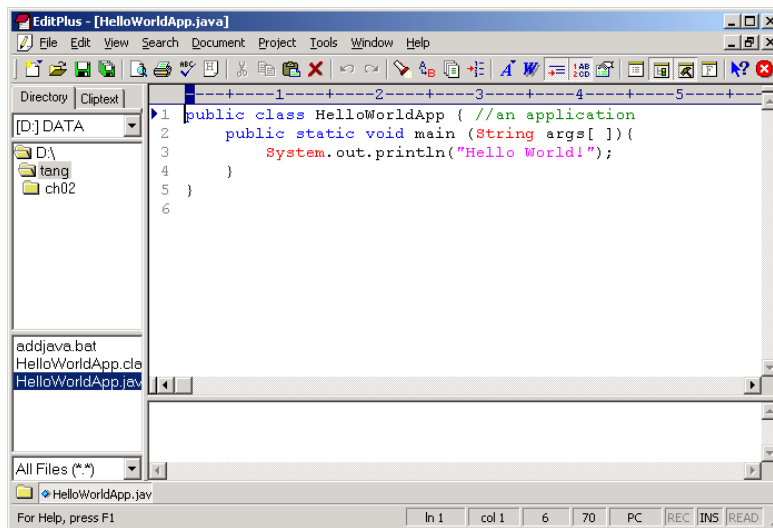


- 常见的三类开发环境
  - ( 1 ) 直接使用JDK
    - Javac, java, appletViewer
  - ( 2 ) 文本工具+调用JDK命令
    - SublimeText, Notepad++, UltraEdit , EditPlus
    - JCreator, Kawa及Freejava
  - ( 3 ) 集成开发环境 ( IDE: integrated development environment)
    - Eclipse, NetBeans ( 这两款用得很广 )
    - Jdeveloper(Oracle), VisualAge for Java(IBM)
    - BlueJ (用于教学 )

# EditPlus的配置



- 配置外部命令 ( Tools—Preference—Tools—UserTools)



# EditPlus的具体配置



	编 译	运 行
Menu text	Compile Java	Run Java
Command	c:\jdk\bin\javac.exe	c:\jdk\bin\java.exe -classpath .
Argument	\$(FileName)	\$(FileNameNoExt)
Initial directory	\$(FileDir)	\$(FileDir)
Capture output	选择	不选择



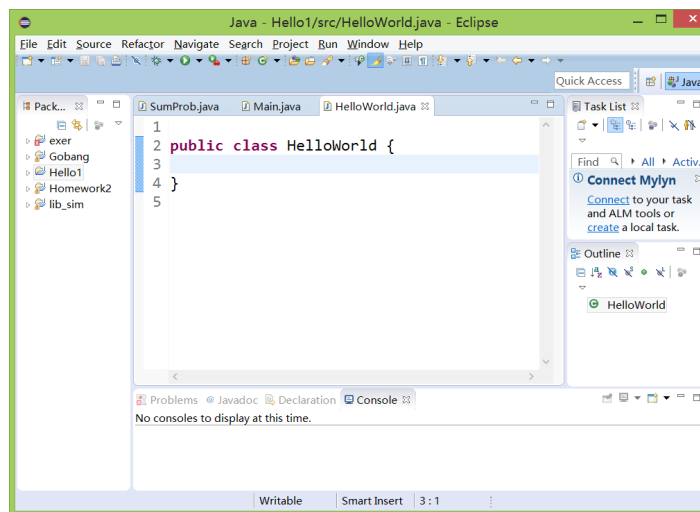
# 常用的集成开发环境(IDE)

- 开源的Eclipse (免费，强烈推荐)
  - 下载：<http://www.eclipse.org/downloads/>
  - 另，Android开发版 <http://developer.android.com/sdk>
- Oracle公司出品的NetBeans(免费，推荐)
  - 下载：<http://java.sun.com>
  - 可以下载与JDK绑定的NetBeans

# 使用Eclipse



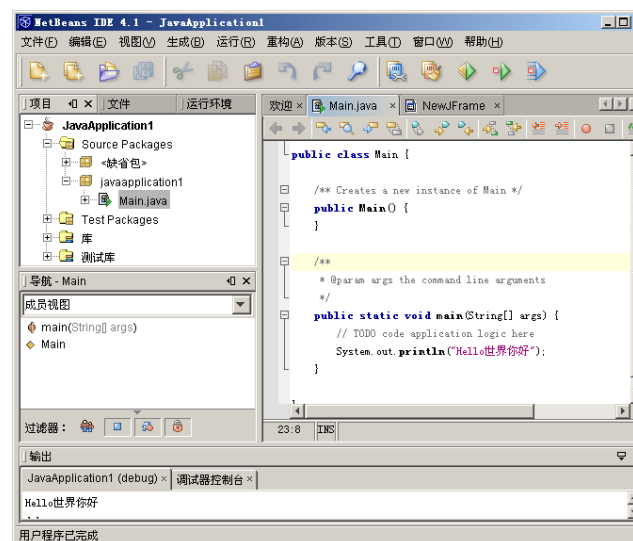
- 与NetBeans相似
- 可以下载多种插件
- 编程
  - 新建一个项目
  - 添加一个类
  - 写上main()方法
  - 使用“运行”命令，或在项目上点右键，Run As ...
  - 如果要打包，则可以使用 File—Export—Java—Runnable jar
  - 如果要生成java doc，则可以使用 Project—Generate Javadoc



# 使用NetBeans



- Project管理器
- 源文件编辑器
- 图形界面构造工具
- Build管理工具
- 调试器



- 在NetBeans IDE中创建一个项目（JavaApplication应用项目）
- 并且在main方法中填写代码，按F6即可以运行该程序
- 或者使用运行菜单中的命令



# IDE中快速输入代码



- 代码模板 (Code Template)
- 在Eclipse中
  - 输入`main`，再按`Alt+/>`键，得到`main`函数
  - 输入`sysout`，再按`Alt+/>`键，即可得到`System.out.println("");`
  - 更多的，可见 Window—Preferences—Java—Editor—Templates
- 在NetBeans中
  - 输入`psvm`，再按`Tab`键，得到`main`函数
  - 输入`sout`，再按`Tab`键，即可得到`System.out.println("");`
  - 更多的，可见 工具—选项—编辑器—代码模板