

Java 入门

1.1 知识提炼

1.1.1 Java 运行平台

随着网络的飞速发展,Java 已经成为网络时代最重要的语言之一,主要优势体现在“一次编写,到处运行(write once, run anywhere)”,这需要提供相应的运行平台,目前 Java 运行平台主要有 3 个版本。

- J2SE(Java 2 Platform Standard Edition)包含构成 Java 语言核心的类,如数据库连接、接口定义、输入/输出和网络编程。主要用于开发一般个人计算机上的应用软件,后更名为 Java SE。
- J2ME(Java 2 Platform Micro Edition)包含 J2SE 中一部分类,主要用于消费类电子产品的软件开发,如呼机、智能卡、手机、PDA 和机顶盒,后更名为 Java SE。
- J2EE(Java 2 Platform Enterprise Edition)包含 J2SE 中的所有类,并且还包含用于开发企业级应用的类,如 EJB、Servlet、JSP、XML 和事务控制,也是现在 Java 应用的主要方向。主要用于开发企业级应用软件,后更名为 Java SE。

1.1.2 Java 程序的运行机制

在 Java 中源文件名称的后缀为.java,之后通过编译生成一个扩展名为.class 的与平台无关的字节码 class 文件,然后由 Java 虚拟机(JVM)解释执行。基本原理如图 1-1 所示。

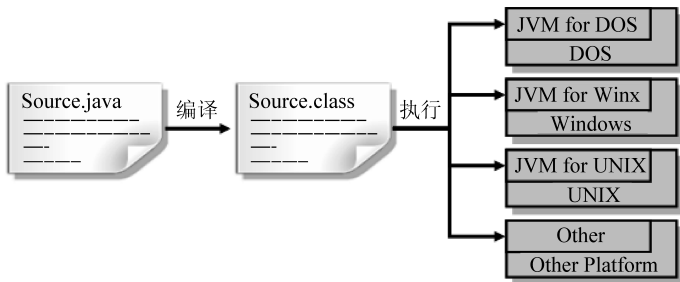


图 1-1 JVM 基本原理

在 Java 中源文件名称的后缀为.java,之后通过编译使 *.java 的文件生成一个 *.class 文件,在计算机上执行 *.class 文件,但是此时执行 *.class 的计算机并不是一个物理上可



以看到的计算机,而是 Java 自己设计的一个计算机——JVM,所有的 *.class 文件都是在 JVM 上运行的,即 *.class 文件只需要认识 JVM,由 JVM 再去适应各个操作系统。如果不同的操作系统安装上符合其类型的 JVM,那么以后程序无论到哪个操作系统上都是可以正确执行的。Java 也是通过 JVM 进行可移植性操作的。

1.1.3 Java SE 开发环境

一台计算机上安装了 JVM 即可运行 Java 程序,但是要开发 Java 程序,还需建立 Java 开发环境。不同领域的 Java 开发应用所需的版本不同,本书使用 Java SE 的开发环境。安装与配置开发环境的步骤如下。

步骤一:安装 JDK(本书使用 JDK 1.7 版本)。

下载安装文件,按照安装向导提示一步步安装完成即可。

步骤二:设置环境变量。

在 JDK 安装完毕需设置 path 和 classpath 两个环境变量,这两个环境变量的设置非常关键,是程序编译和运行的重要保证。path 指示 java 命令的路径,像 javac、java、javaw 等,这样在控制台下面编译、执行程序时就不需要再输入具体路径了。classpath 是类库的默认搜索路径,即告诉 JVM 要使用或者执行的 *.class 文件所在的目录。这个是专门针对 Java 的,故系统里没有这条路径,即告诉 JVM 要使用或者执行的 *.class 文件所在的目录。由于 JDK 的安装路径多次使用,在此先新建环境变量 JAVA_HOME。

(1) 新建环境变量 JAVA_HOME,其值为 C:\Java\jdk1.7.0_03,如图 1-2 所示。

(2) 设置变量 Path,在变量值最前面增加 %JAVA_HOME%\bin;,如图 1-3 所示。



图 1-2 设置 JAVA_HOME 路径

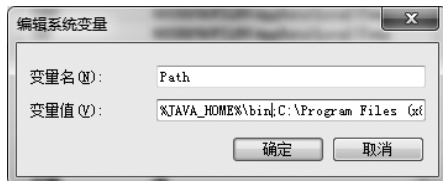


图 1-3 设置 Path 路径

注意: 环境变量的各变量值之间需用分号分隔。

(3) 新建变量 classpath,其值为.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar,如图 1-4 所示。注意最前面是“.”。

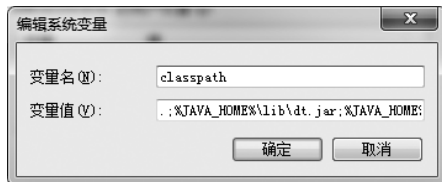


图 1-4 设置 classpath 路径



1.1.4 Java 开发工具

Java 的开发工具很多,目前比较流行的 Java 开发工具有 EditPlus、Jcreator、Eclipse、MyEclipse、Jbuilder、NetBeans 等,本书主要用的开发工具是 MyEclipse。MyEclipse 的功能非常强大,支持也十分广泛,尤其是对各种开源产品的支持。MyEclipse 可以支持 Java Servlet、AJAX、JSP、JSF、Struts、Spring、Hibernate 及 EJB3、JDBC 数据库链接工具等多项功能。图 1-5 为 MyEclipse 工作界面。

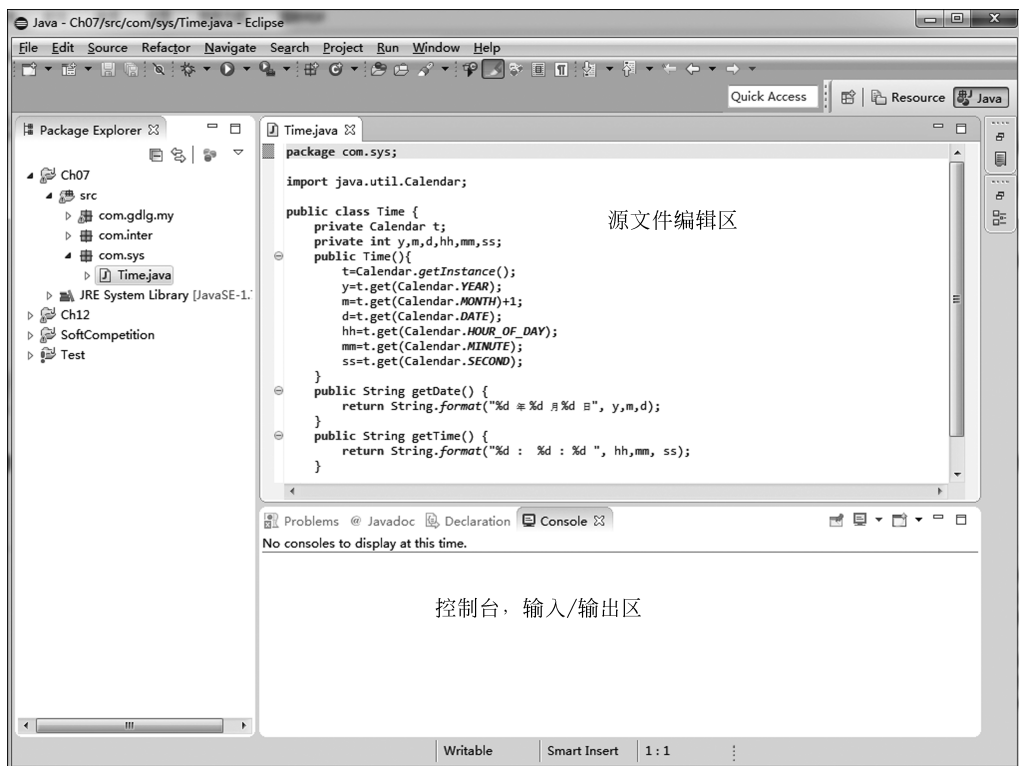


图 1-5 MyEclipse 工作界面

常用操作如下。

(1) 新建 Java Project: File→New→Java Project,如图 1-6 和图 1-7 所示。

(2) 新建各类对象:包(Package)、类(Class)等。

① 新建类的操作:选中相应的 Project 的 src 目录,右击 New→Class,如图 1-8 和图 1-9 所示。

② 新建包的操作:选中相应的 Project 的 src 目录,右击 New→Package,如图 1-10 所示,输入相应的包的名称即可,注意包的名称用小写字母,可以用点作为分隔符。本实验的代码均放在 gdlgxy.shiyan1 包中。

(3) 如果有些窗口不小心关闭了,可以使用 Show View 功能,如图 1-11 所示。

(4) 代码格式化,如图 1-12 所示,使用快捷键 Ctrl+Shift+F。

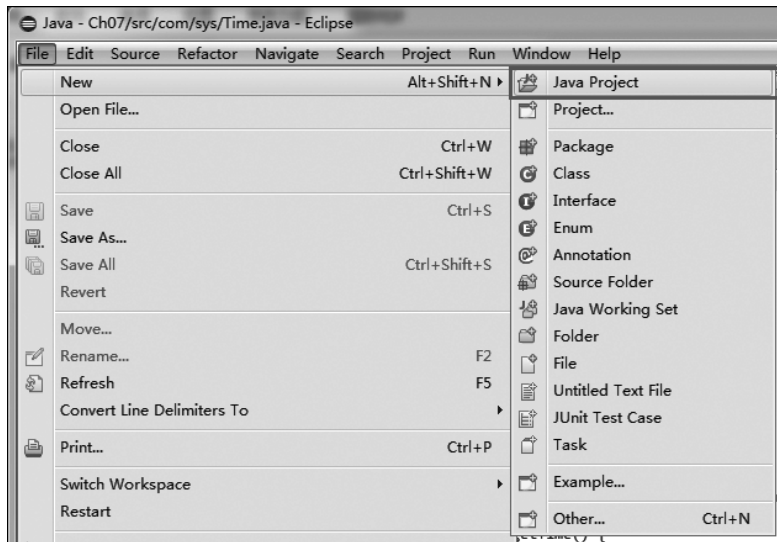


图 1-6 新建 Project(1)

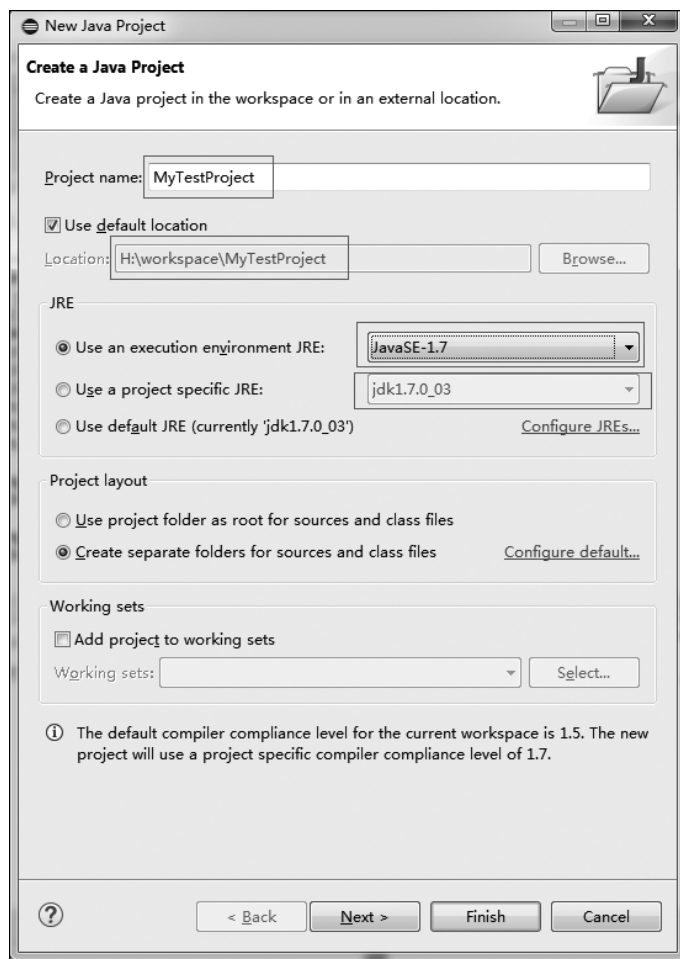


图 1-7 新建 Project(2)

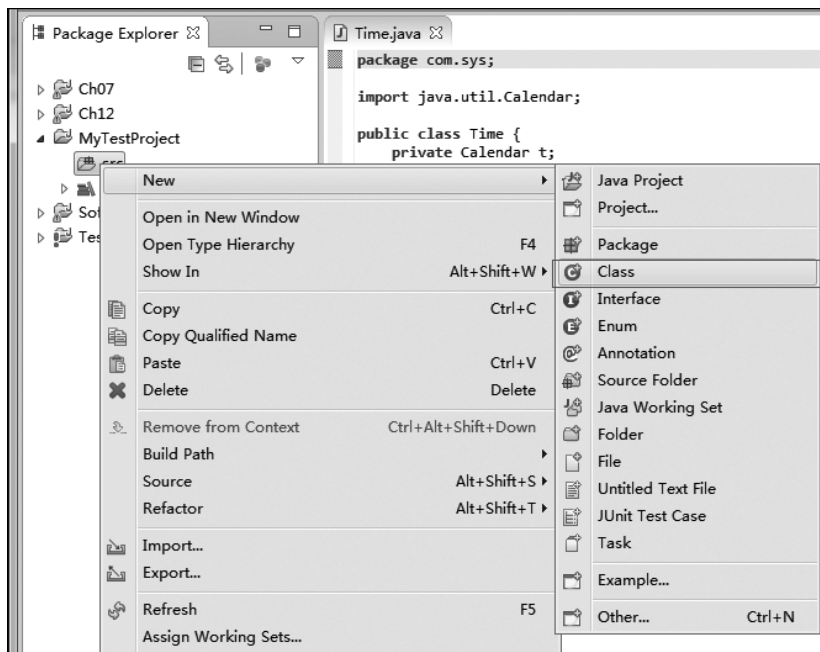


图 1-8 新建 Class(1)

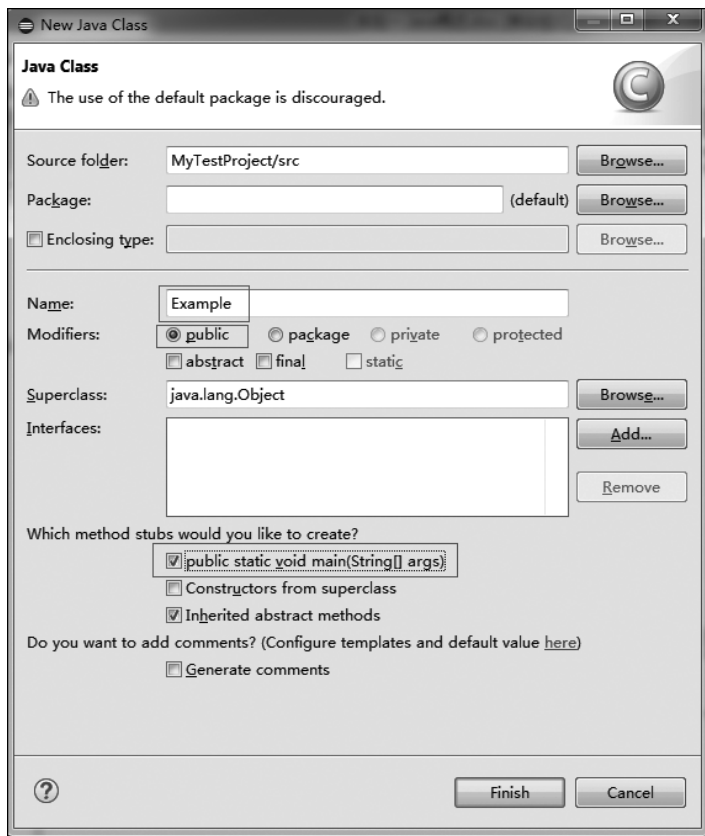


图 1-9 新建 Class(2)

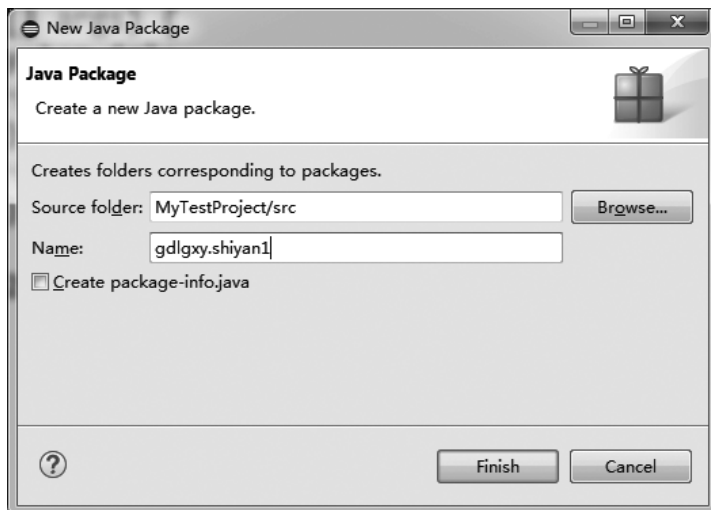


图 1-10 新建包

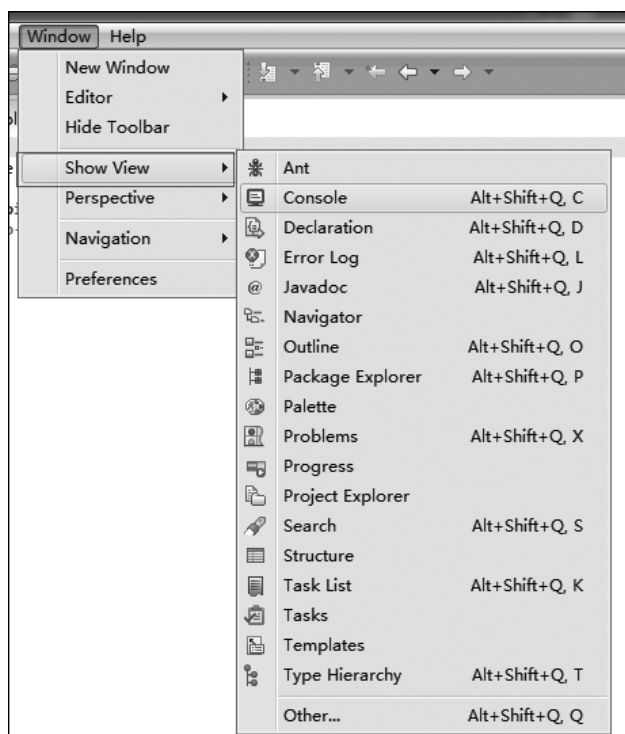


图 1-11 Show View

(5) 重命名: 选中需重命名的对象, 右击 Refactor→Rename..., 如图 1-13 和图 1-14 所示。在图 1-14 所示界面中输入新名称即可。

(6) 修改编辑区文本字体, 如图 1-15 所示。

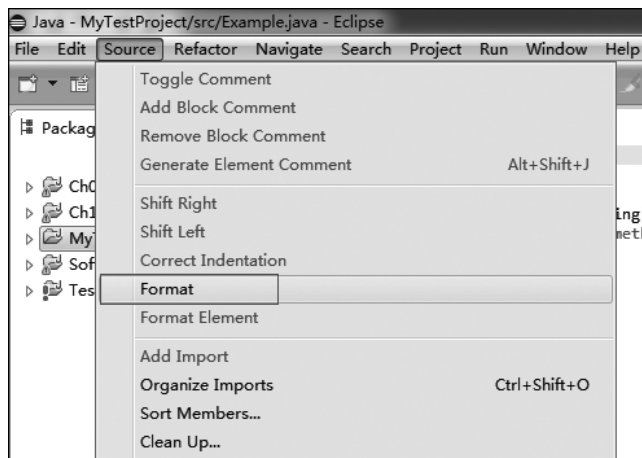


图 1-12 代码格式化

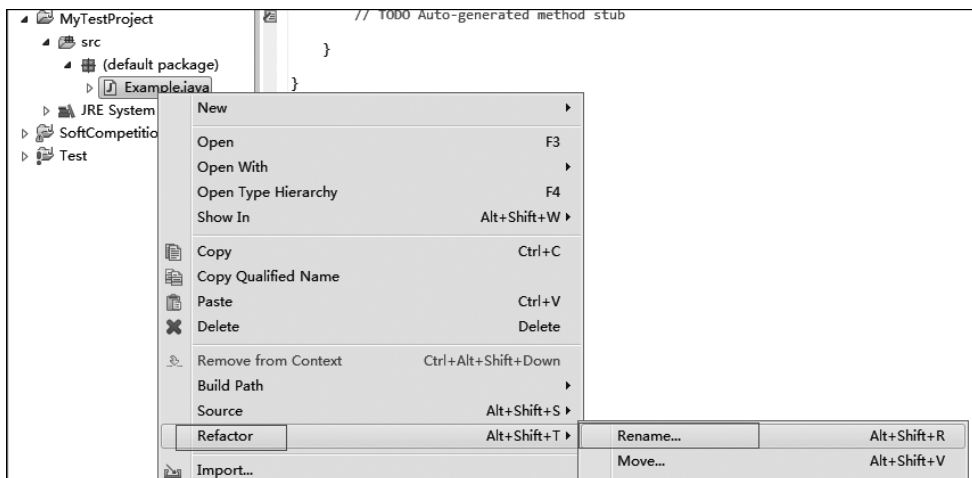


图 1-13 重命名(1)

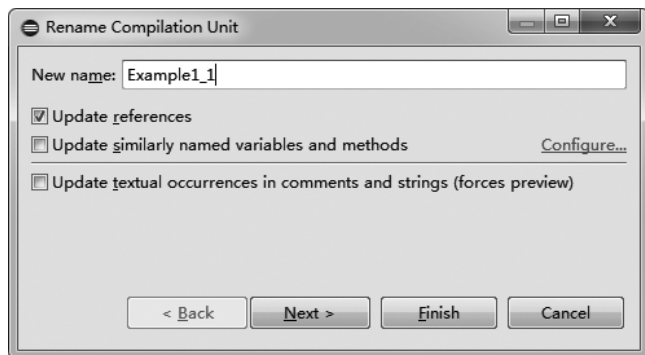


图 1-14 重命名(2)

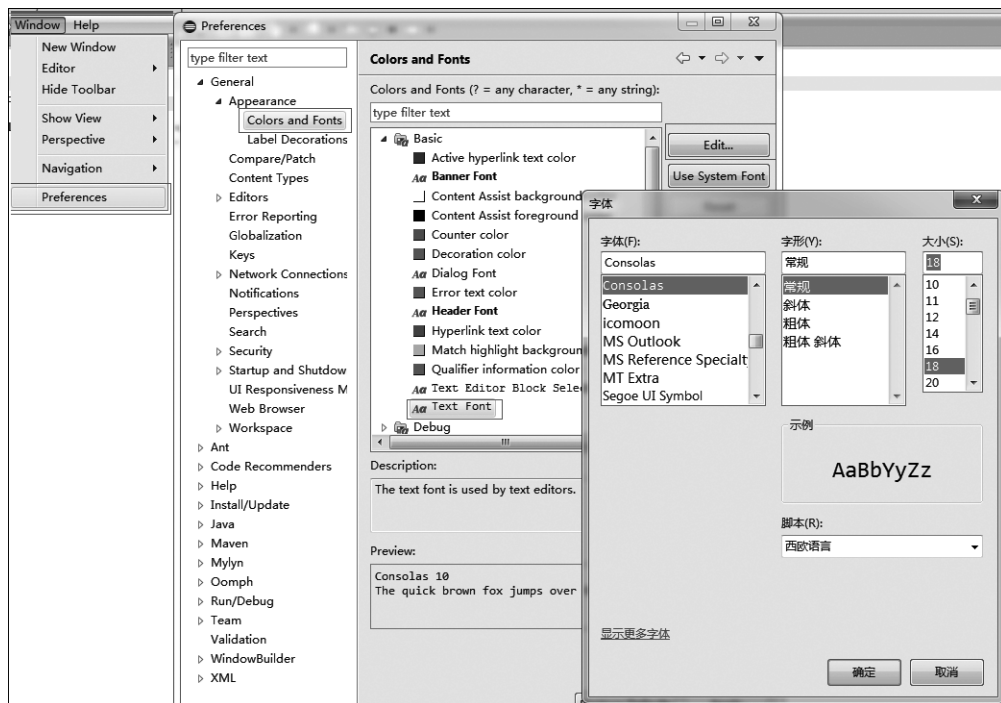


图 1-15 修改文本字体

1.1.5 Java 应用程序开发过程

Java 应用程序的开发过程是编写源文件(扩展名为.java)→编译源文件→运行应用程序,如图 1-16 所示。

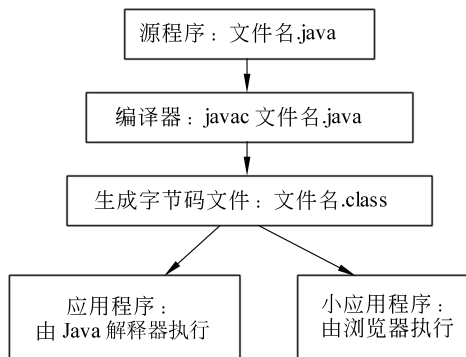


图 1-16 Java 程序的开发过程

Java 程序的基本结构是类,一个源文件里可以只有一个类,通常情况下源文件名即为类名,编译源文件得到这个类的字节码文件。一个源文件中也可以包含多个类,其中最多只能有一个 public 类,如果这个源文件中有一个 public 类,那么源文件名必须为 public 类的类名,编译源文件后会得到多个字节码文件,即每个类会生成一个字节码文件。



1.2 实例解析

【实例要求】

编写一个简单问答程序：系统提示输入姓名，用户输入姓名，系统输出欢迎信息，如图 1-17 所示。

请输入您的姓名：张三丰
张三丰欢迎您！
祝您学有所成！

【实例源代码】

图 1-17 简单问答程序

新建源文件 ShiyanDemol.java，源代码如下：

```
package gdlgxy.shiyan;
import java.util.Scanner;           //导入 Scanner 类

public class ShiyanDemol {
    /* 功能描述：系统提示输入姓名，用户输入姓名，系统输出欢迎信息
    */
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);    //构造 Scanner 类对象 sc
        System.out.print("请输入您的姓名："); //提示输入姓名
        String name=sc.next();                //接收，用户输入赋值给字符串变量 name
        System.out.println(name+"欢迎您！\n 祝您学有所成！"); //输出欢迎信息
    }
}
```

【实例解析】

(1) MyEclipse 的使用方法：新建 Java 项目、新建包、新建类。

(2) 读入数据。

在 java.util 包中，可使用 Scanner 类创建一个对象，实现数据的输入。因此，在前面需要用到 import 语句导入 Scanner 类。读入数据的方法有：

- nextInt()——读入整数。
- nextFloat()——读入浮点数。
- next()、nextLine()——读入字符串。

注意：next()一定要读取到有效字符后才结束输入，对输入有效字符之前遇到的空格键、Tab 键或 Enter 键等结束符，next()方法会自动将其去掉，只有在输入有效字符之后，next()方法才将其后输入的空格键、Tab 键或 Enter 键等视为分隔符或结束符。next()查找并返回再次扫描下一个完整标记。完整标记的前后是与分隔模式匹配的输入信息，所以 next()方法不能得到带空格的字符串。而 nextLine()方法的结束符只是 Enter 键，即 nextLine()方法返回的是 Enter 键之前的所有字符，它是可以得到带空格的字符串的。



(3) 输出数据。

- println()方法输出信息后换行。

```
System.out.println();           //输出换行符
System.out.println("XXX");      //输出信息并换行
```

- print()方法输出信息不换行。

```
System.out.print("XXX");
```

- printf()方法格式化输出。

```
System.out.printf("格式控制符", 参数 1, 参数 2, ..., 参数 n);    //格式化输出
```

其中,格式控制符字符串由普通字符和格式控制符组成,普通字符原样输出,格式控制符用以控制后面的参数以何种格式输出,后面的参数个数与格式控制符格式一致。格式控制符如下。

%d: 输出 int 型数据。

%md: 输出占 m 列 int 型数据。

%f: 输出 float、double 浮点数。

%.nf: 输出小数保留 n 位的浮点数。

%m.nf: 输出占 m 列小数保留 n 位的浮点数。

%e: 以指数形式输出 float、double 浮点数。

%c: 输出 char 型数据。

%s: 输出 String 型数据。

(4) 养成良好的编程习惯,如缩进、写注释等,掌握单行注释和多行注释方法。

1.3 上机实验

【实验目的】

- 了解常用的 Java 开发工具。
- 掌握安装并配置 Java 开发环境。
- 掌握 Java 的开发流程。
- 可以编写并运行一个简单的 Java 程序。

【实验要求】

使用 Java 语言完成一个简单的个性测试小程序。

依次提出以下问题,计算机给出判定回答。

你走到森林里,希望第一眼看见什么东西?

你走到森林里,希望第二眼看见什么东西?

继续往前走,看见一个屋子,你是绕一圈进去还是直接推门进去? 答案请填写直接或者绕弯。