

《JAVA 技术》课程实验指导书

西安电子科技大学软件学院

撰写人：刘惠

2016. 1

目 录

第一章 JAVA 开发环境及开发工具介绍	4
1.1 JDK 安装与配置	4
1.2 Eclipse 开发工具介绍	10
实验一 JAVA 基础知识编程实验	17
1.1 基础知识部分	17
1.1.1 实验目的	17
1.1.2 实验内容	17
1.1.3 实验示例	错误！未定义书签。
1.1.4 实验思考	17
1.1.5 实验总结	18
1.2 面向对象编程	18
1.2.1 知识准备	18
1.2.2 实验目的	21
1.2.3 实验内容	22
1.2.4 实验示例	错误！未定义书签。
1.2.5 实验思考	23
1.2.6 实验总结	23
实验二 JAVA GUI 编程实验	错误！未定义书签。
2.1 编写一个带界面的计算器程序	错误！未定义书签。
2.1.1. 知识准备	错误！未定义书签。
2.1.2 实验目的	错误！未定义书签。
2.1.3 实验内容	错误！未定义书签。
2.1.4 实验示例	错误！未定义书签。
2.1.5 实验思考	错误！未定义书签。
2.1.6 实验总结	错误！未定义书签。
2.2 仿 QQ,编写聊天室客户端界面	错误！未定义书签。
2.2.1. 实验目的	错误！未定义书签。

2.2.2 实验内容	错误！未定义书签。
2.2.3 实验示例	错误！未定义书签。
2.2.4 实验思考	错误！未定义书签。
2.2.5 实验总结	错误！未定义书签。
实验三 JAVA IO 操作和网络编程实验	24
3.1 编写 IO 文件处理程序	错误！未定义书签。
3.1.1 知识介绍	错误！未定义书签。
3.1.2 实验目的	错误！未定义书签。
3.1.3 实验内容	错误！未定义书签。
3.1.4 实验示例	错误！未定义书签。
3.1.5 思考	错误！未定义书签。
3.1.6 实验总结	错误！未定义书签。
实验四 JAVA 数据库操作（JDBC）	错误！未定义书签。
4.1 编写 Mysql 数据库的插入和查询数据程序	错误！未定义书签。
4.1.1 知识介绍	错误！未定义书签。
4.1.2 实验目的	错误！未定义书签。
4.1.3 实验内容	错误！未定义书签。
4.1.4 实验示例	错误！未定义书签。
4.1.5 思考	错误！未定义书签。
4.1.6 实验总结	错误！未定义书签。
实验五 综合大作业：C/S 架构的聊天室设计与实现	27
5.1 完成网络聊天室应用	27
5.1.1 知识介绍	27
5.1.2 实验目的	28
5.1.3 实验内容	28
5.1.4 实验示例	28
5.1.5 思考	35
5.1.6 实验总结	35

第一章 JAVA 开发环境及开发工具介绍

1.1 JDK 安装与配置

1.1.1 下载安装 jdk

JDK 开发工具包可以免费下载。下载网址——
<http://java.sun.com/product>

下载如下文件：jdk-1_5_0-windows-i586.exe。

由于目前大多数用户使用的是 Window 操作系统，所以我们在 Window 操作系统上安装 jdk-1_5_0-windows-i586.exe 为例，说明安装 jdk-1_5_0 的过程。

jdk-1_5_0-windows-i586.exe 是一个自解压文件，双击它就可以解压缩，同时进行安装工作，双击 jdk-1_5_0-windows-i586.exe，文件会自动解压缩，如图 2.1 所示。

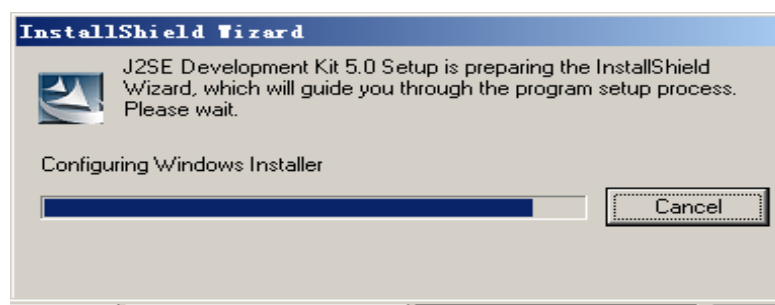


图 1.1 jdk-1_5_0-windows-i586.exe 的解压过程

解压缩工作完成之后，会出现 JDK1.5.0 的许可协议，如图 1.2 所示

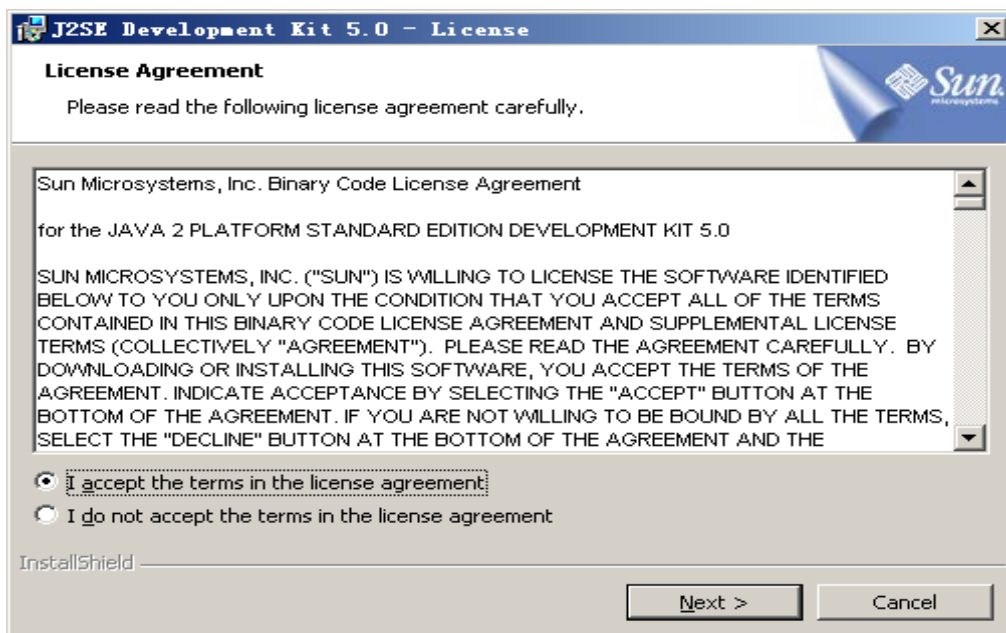


图 1.2 选择许可协议界面

点击“next”按钮，出现如图 1.3 所示对话框，用户可选择要安装的组件及选择安装目录。点击“change”，可改变安装目录。

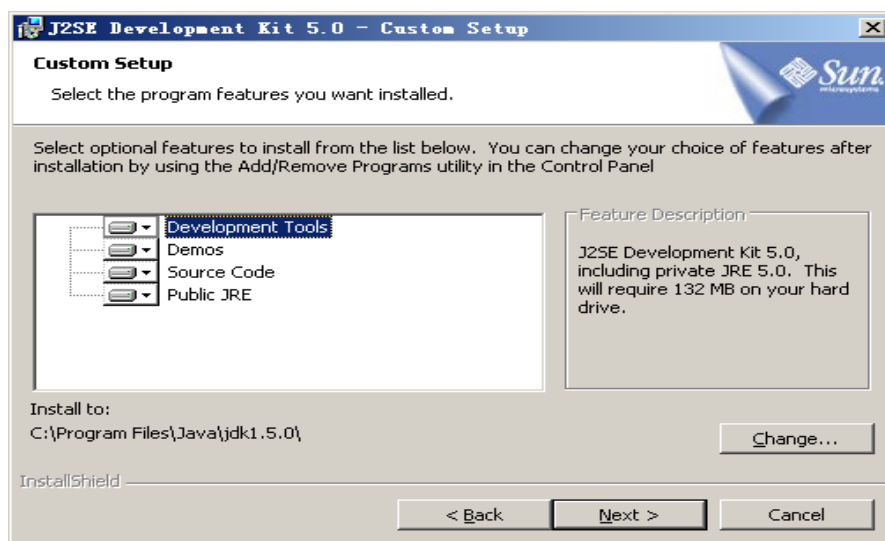


图 1.3 组件的选择

选好后，点击“next”按钮，开始安装，如图 1.4 所示。

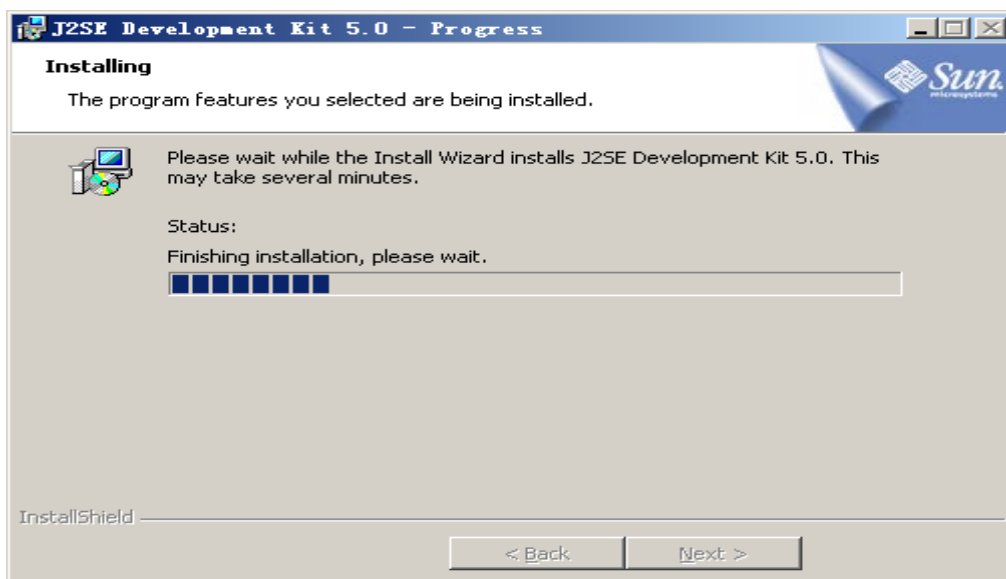


图 1.4 安装进度界面

运行环境设置及安装，如图 1.5 所示。

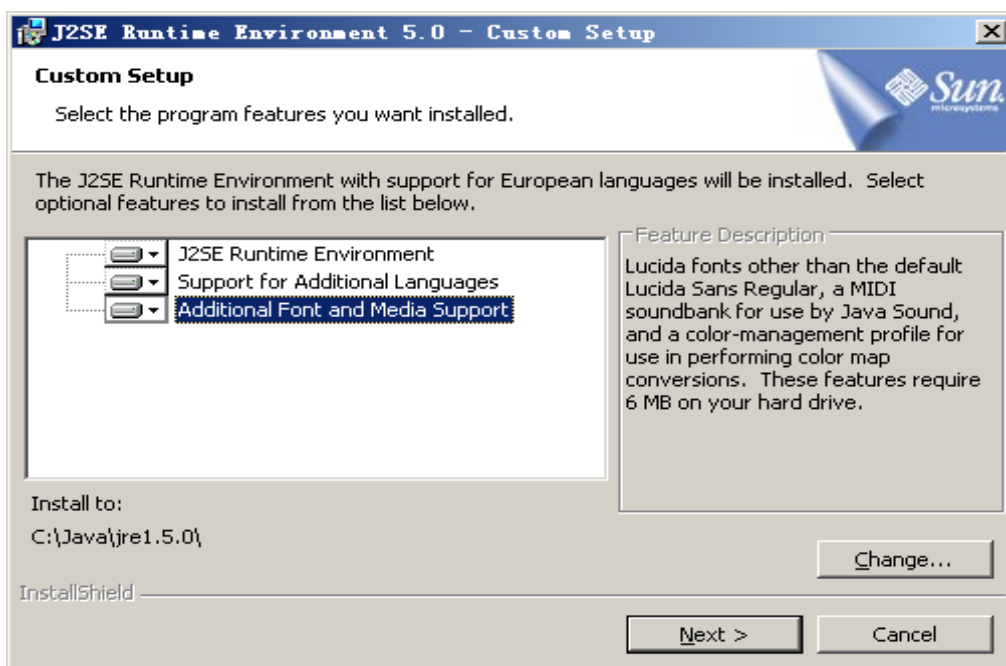


图 1.5 安装过程界面

单击“Next”按钮，出现如图 1.6 所示对话框，选择浏览器，选中后，单击“Next”按钮继续安装。

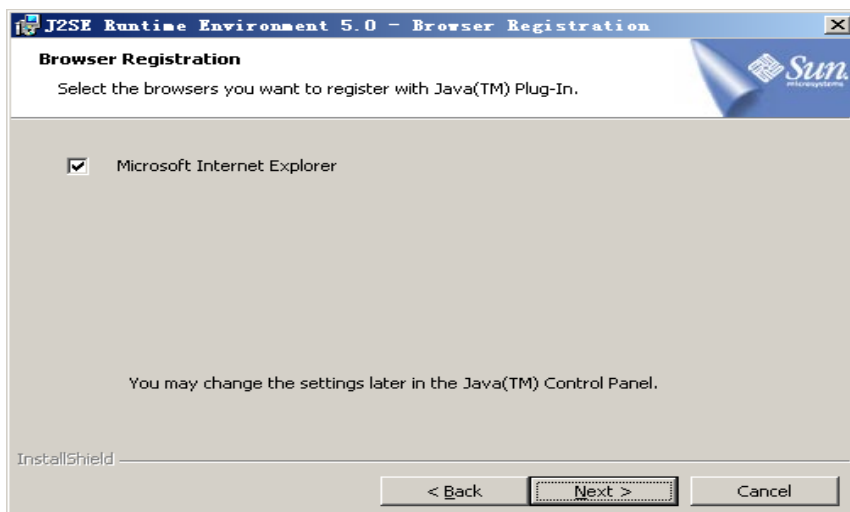


图 1.6 选择浏览器界面

最后出现如图 1.7 所示界面，单击“finish”按钮结束安装。

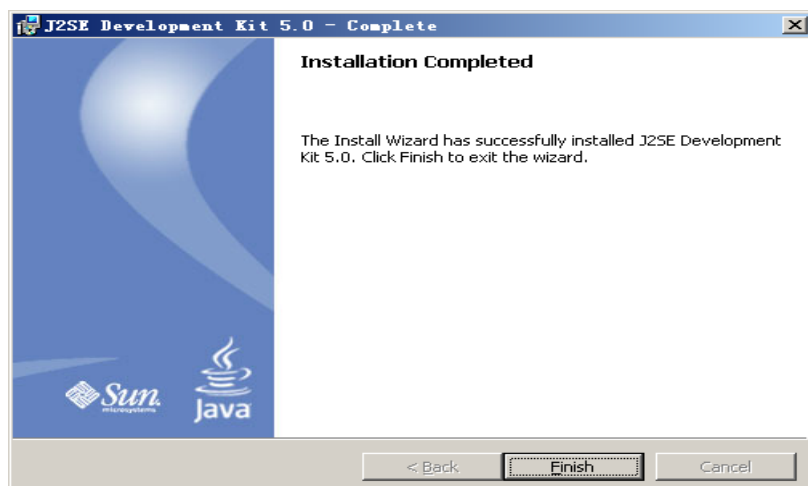


图 1.7 安装结束界面

1.1.2 配置环境变量

1. 方法一

平台为 Windows 2000 时，需要修改系统根目录下的 autoexec.bat 文件。使用记事本或其它文本编辑器打开文件 autuexec.bat，在该文件的最后增加如下两行

```
SET PATH=%CLASSPATH%;. ;C:\Program Files\Java\jdk1.5.0_04\jre\lib\rt.jar
```

```
SET PATH=%PATH%;C:\Program Files\Java\jdk1.5.0_04\bin;
```

其中，环境变量 `path` 和 `classpath` 分别指定了 JDK 命令搜索路径和 Java 类路径。在这里我们假设 JDK 安装在 `C:\Program Files\Java\jdk1.5.0_04` 目录下，JDK 的所有命令都放在 `C:\Program Files\Java\jdk1.5.0_04\bin` 目录下。设置环境变量 `path` 的作用是使 DOS 操作系统可以找到 JDK 命令。设置环境变量 `classpath` 的作用是告诉 Java 类装载器到哪里去寻找第三方提供的类和用户定义的类。JVM 和其它 JDK 工具通过依次搜索平台库、库扩展和类路径来查找类。

如果用户在安装 `jdk1.5.0` 时，选择了另外的 JDK 安装路径，则环境变量 `path` 和 `classpath` 要作相应的调整。

2. 方法二

右单击“我的电脑”，选择“属性”，出现如图 1.7 对话框，选择“高级”选项卡，选择其中的“环境变量”，出现如图 1.8 对话框，在系统变量中建立或设置环境变量。在“变量名”对话框中输入环境变量名，在“变量值”对话框中输入相关值得路径。设置过程如图 1.9、图 1.10 所示，设置完成后，单击“确定”完成设置。

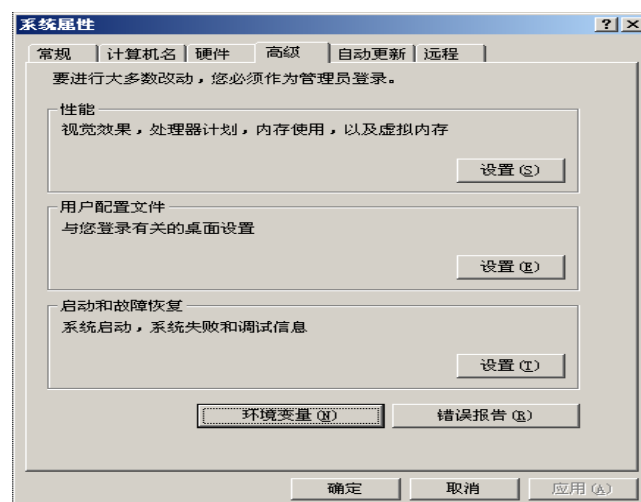


图 1.7 系统属性对话框

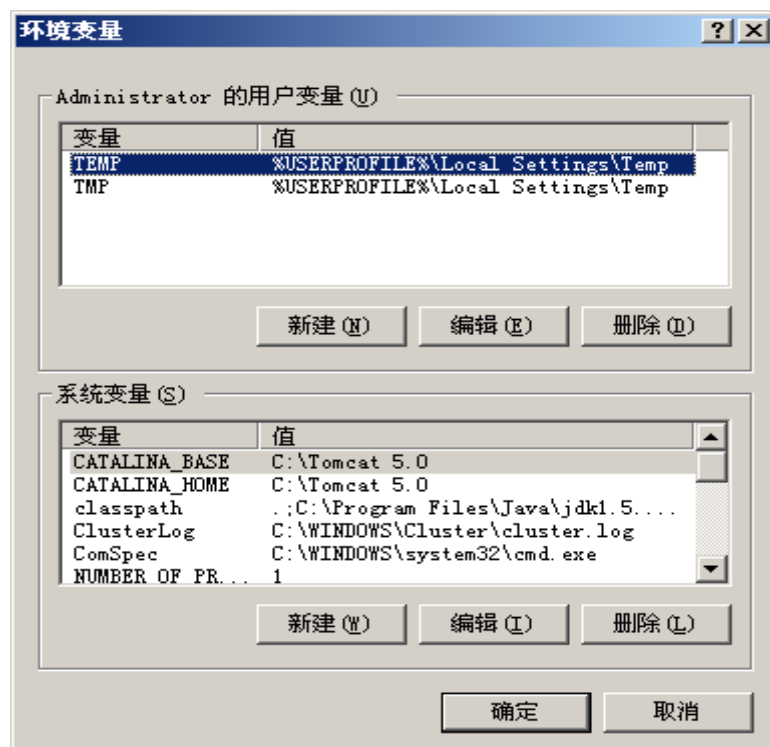


图 1.8 设置环境变量

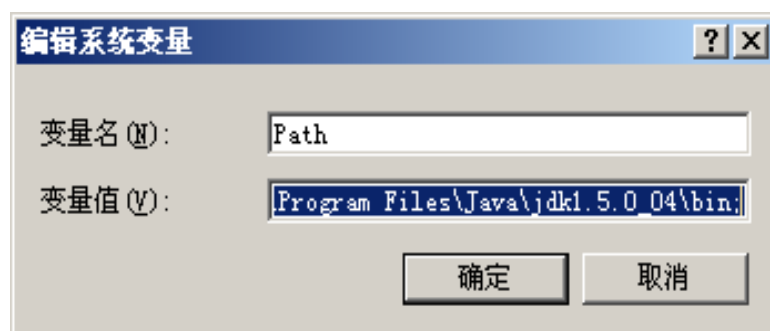


图 1.9 新建环境变量 path

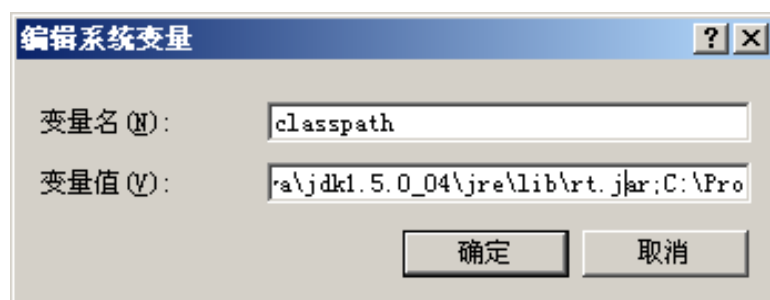


图 1.10 新建环境变量 classpath

1.1.3 测试安装

环境变量设置完成后,重新启动计算机,在 DOS 窗口下,敲入 javac

并回车后，如果出现 Javac 的用法参数提示信息，则安装正确。否则要检查环境变量设置是否正确。

1.2 Eclipse 开发工具介绍

1.2.1 Eclipse 的安装

使用 Eclipse IDE 软件可以很方便地进行编译和运行程序的工作，但如果程序有交互的功能，还需要安装其它插件。

将文件 eclipse-SDK-3.2-win32.zip 拷贝到 C: 盘根目录下，然后右单击该文件将其解压到当前文件夹下，如图 1.11 所示。解压后的目录结构如图 1.12 所示。

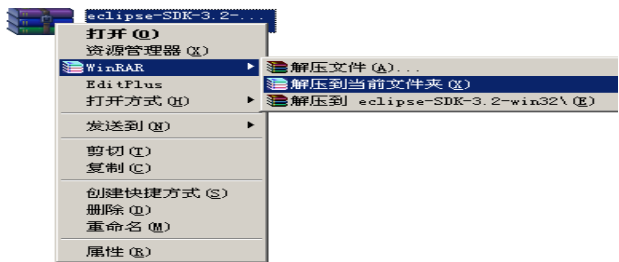


图 1.11 解压操作界面

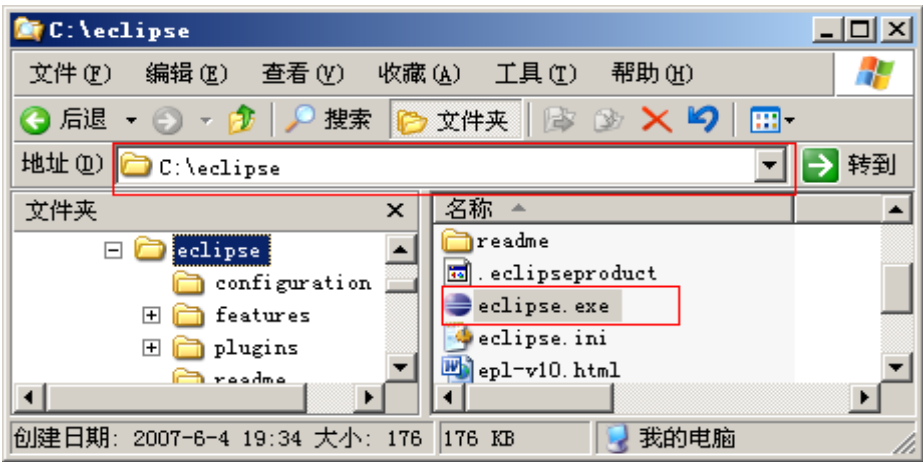


图 1.12 解压后的目录结构图

双击 C:\eclipse\eclipse.exe 文件，可看到如图 1.13 所示启动界面。

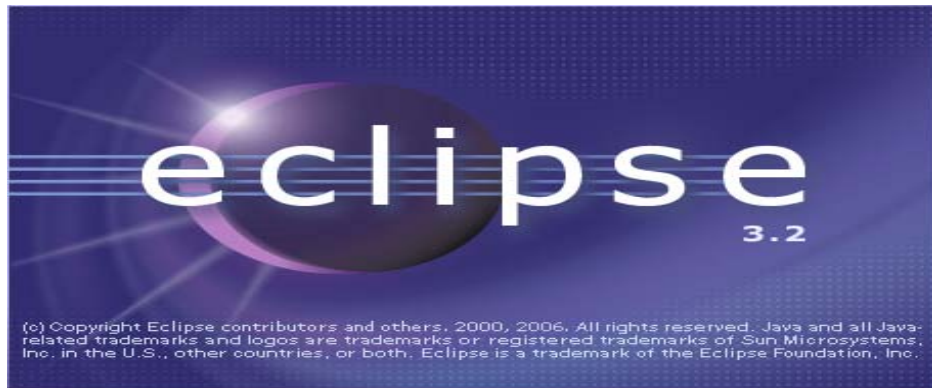


图 1.13 eclipse 启动界面

改变工作目录如图 1.14 所示。

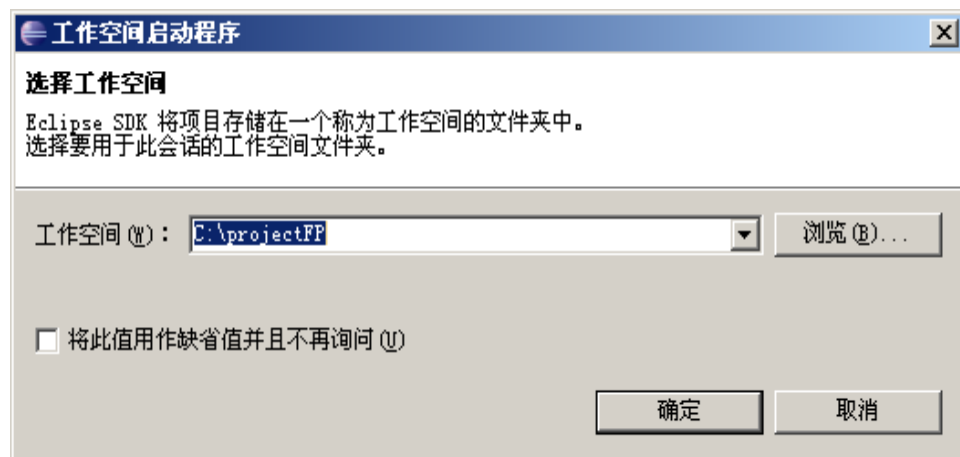


图 1.14 选择工作区界面

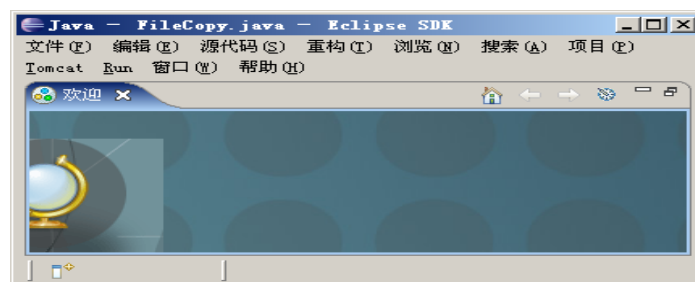


图 1.15 选择工作区界面

选择好工作空间以后，单击“确定”按钮，如果能看到如图 1.15 所示界面，则安装成功。

1.2.2 使用 Eclipse

在 jdk1.5.0 的 bin 目录下，存放着 Java 2 提供的一些可执行程序，为我们开发和测试 Java 程序提供了工具。在我们的学习中，常用的 JDK 开发工具有如下几种：

- (1) javac.exe: Java 语言的编译器。
- (2) java.exe: Java 程序执行引擎。
- (3) appletviewer: 小应用程序浏览器。

在使用 Eclipse 之前，必须熟练掌握这些工具的用法。

单击“文件 ➤ 新建 ➤ 项目 ...”，如图 2.1 所示。

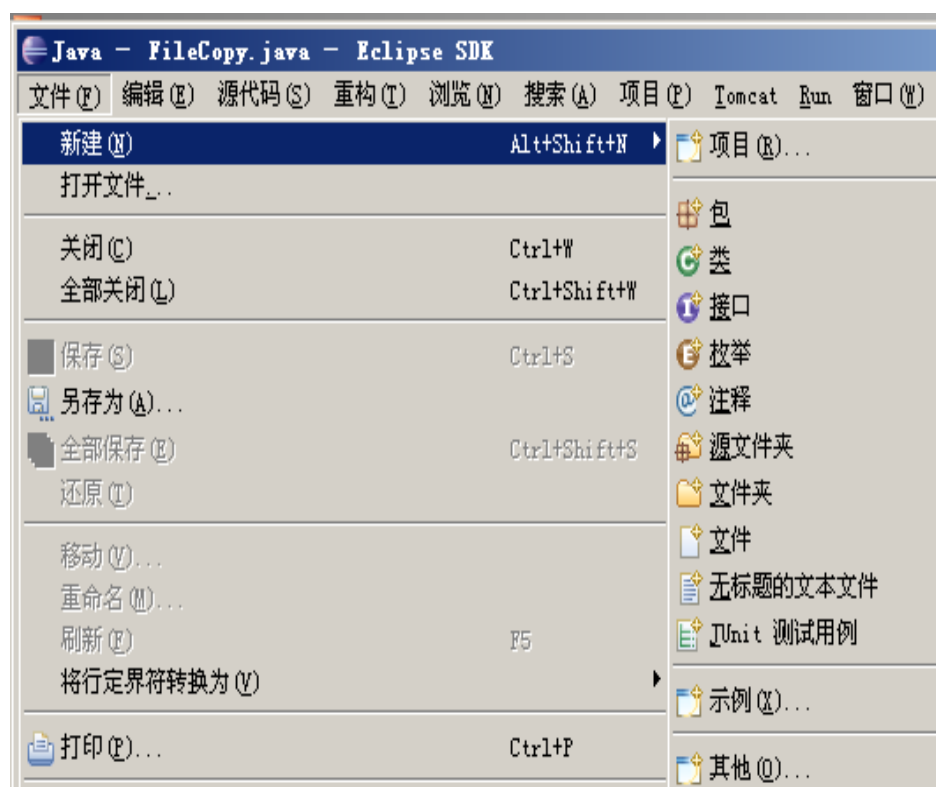


图 2.1 新建项目界面

选择了项目后出现如下界面，选择“Java 项目”后单击“下一步”。



图 2.2 新建 Java 项目界面

在图 2.3 所示界面中，在“项目名”一栏中，填入“Simple Java Program”其它使用缺省值。单击下一步，出现如图 2.4 所示界面。



图 2.3 新建 Java 项目界面

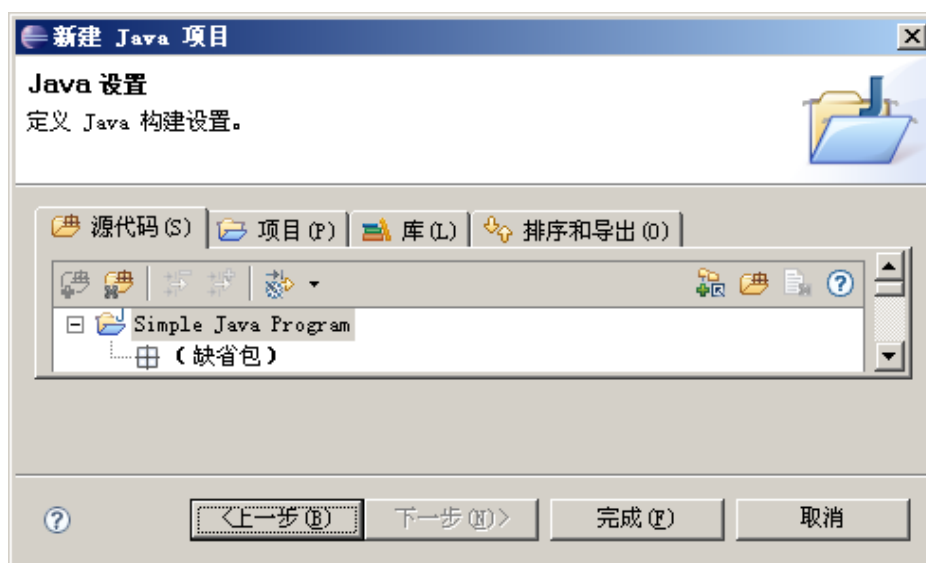


图 2.4 新建 Java 项目界面

在如图 2.4 所示界面中单击“库”选项卡，出现如图 2.5 所示界面，确保 JRE 设置没问题。单击“完成”按钮。



图 2.5 新建 Java 项目界面

在包资源管理器中右单击“Simple Java Program”项目如图 2.6 所示，点击“新建 ➤ 类 ...”，弹出如图 2.7 所示界面。



图 2.6 新建 Java 类界面

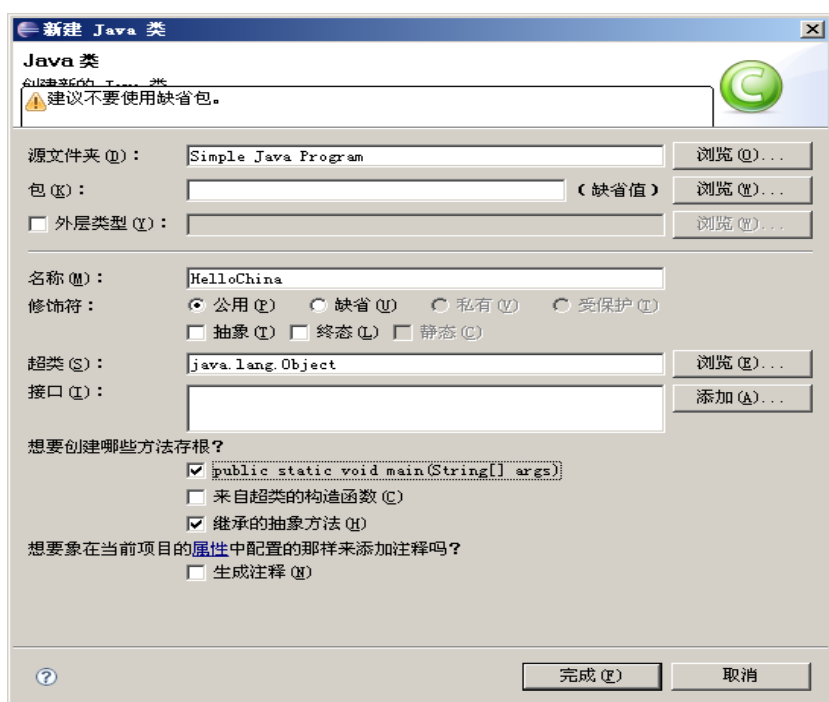


图 2.7 新建 Java 类界面

在图 2.7 所示界面中在“名称”一栏中填写“HelloChina”,其它设置如图 2.7 所示。单击“完成”。在包资源管理器中查看源程序如图 2.8 及图 2.9 所示。



图 2.8 包资源管理器中结构图

```

public class HelloChina {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO 自动生成方法存根
        System.out.println("Hello,China!");
    }
}

```

图 2.9 代码视图

在包资源管理器中右单击“HelloChina.java”,如图 2.10 所示点击“运行方式 ➤ Java 应用程序 …”,运行结果如图 2.11 所示。

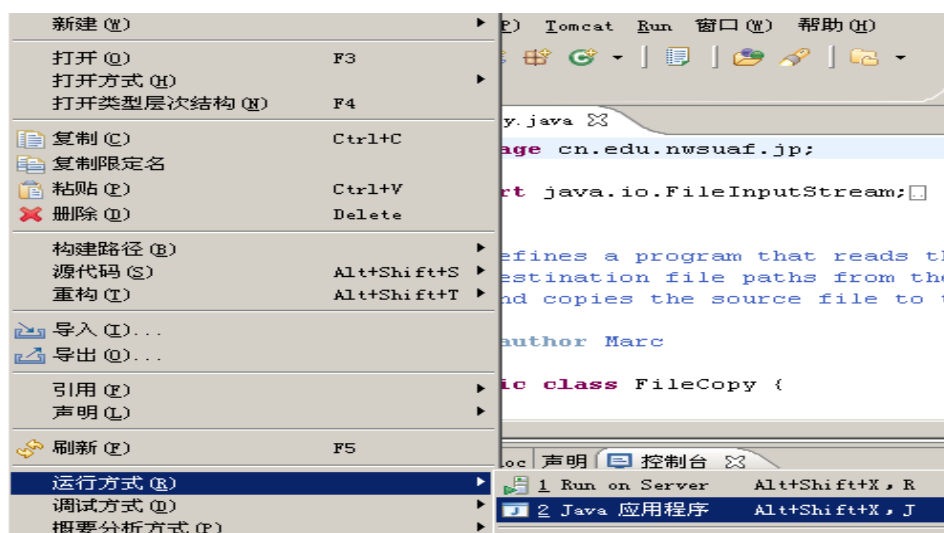


图 2.10 运行界面

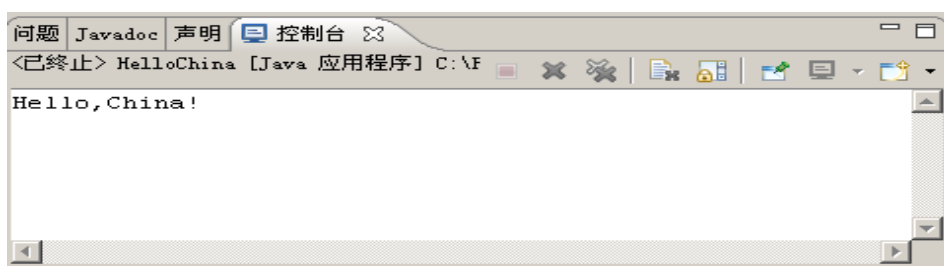


图 2.11 运行界面

实验一 JAVA 基础知识编程实验

1.1 基础知识部分

1.1.1 实验目的

- (1) 掌握使用 JDK 开发 Java 程序的步骤。掌握 Java 程序的基本语法;
- (2) 熟练运用 java 的三个工具编译或运行程序。

1.1.2 实验内容

内容一: 用 ECLIPSE 开发实现 HelloWorld! 程序。

内容二: 实现 JAVA 版本的快速排序。

1.1.4 实验思考

- (1) 在 JAVA 语言没有指针的情况下,编程方面与有指针的语言有何不同?各有什么优势劣势?
- (2) 定义链表类, 实现单链表基本操作。

```
public class Node {  
    public int data;  
    Node next;  
    public Node(int data) {  
        this.data = data;  
        this.next=null;  
    }  
    public void NodeDisplay(){
```

```

        System.out.println ("{" +data+"}");
    }
}

public class Link{
    private Node first;

    public Link () {
        this.first=null;
    }

    public boolean isEmpty(){    }

    public void insertHeadNode(int data){    }

    public Node deleteHeadNode(){    }

    public void findNode(int k){    }

    public void displayLink (){    }

}

```

1.1.5 实验总结

列举出在本实验中遇到的难点问题,以及解决问题的思路 and 方案。

1.2 面向对象编程

1.2.1 知识准备

(一) JAVA 语言的面向对象特征:

1.抽象:

抽象——就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。(就是把现实世界中的某一类东西，提取出来，用程序代码表示，抽象出来一般叫做类或者接口。)

抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是数据抽象，二是过程抽象。

数据抽象——就是用代码的形式表示现时世界中一类事物的特性，就是针对对象的属性。比如建立一个鸟这样的类，鸟都有以下属性：一对翅膀、两只脚、羽毛等。抽象出来的类都是鸟的属性，或者成员变量。

过程抽象——就是用代码形式表示现实世界中事物的一系列行为，就是针对对象的行为特征。比如鸟会飞、会叫等。抽象出来的类一般都是鸟的方法。

2.继承：

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。因此可以说，继承是为了重用父类代码，同时为实现多态性作准备。

3.封装：

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。封装隐藏了类的内部实现机制，从而可以在不影响使用者的

前提下改变类的内部结构，同时保护了数据。

4. 多态性：

多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。总的来说，方法的重写、重载与动态链接构成多态性。Java 引入多态的概念原因之一就是弥补类的单继承带来的功能不足。（为规避 C++ 中多继承造成的复杂继承问题，java 采用单继承。）

动态链接——对于父类中定义的方法，如果子类中重写了该方法，那么父类类型的引用将会调用子类中的这个方法，这就是动态链接。

子类对象可以直接当成父类对象使用，但反过来就不可以。举例来说，人是父类，学生是人的子类，所以学生对象一定具备人对象的属性，但是人对象就未必具有学生对象的特性。所以学生对象可以当做人对象来使用，但是人对象就不能当做学生对象使用。注意当把子类对象当成父类对象使用时，子类对象将失去所有的子类特性，只保留与父类同名的属性和方法（同名方法不仅是函数名相同，而且参数类型也要一样，否则不予保留）。此时可以对父类方法进行重写。

一个类中如果定义了重载的方法，则系统在调用方法时，会根据参数的类型自动选择调用合适的方法。

（二）JAVA 类的访问控制：

类成员访问修饰符与访问能力之间的关系（类成员访问权限）

类型	Private	无修饰 (friendly)	Protected	Public
同一类	可访问	可访问	可访问	可访问
同一包中的子类	不可访问	可访问	可访问	可访问
同一包中的非子类	不可访问	可访问	可访问	可访问
不同包中的子类	不可访问	不可访问	可访问	可访问
不同包中的非子类	不可访问	不可访问	不可访问	可访问

（三）JAVA 中不同控制符对应的权限

继承和权限：

	Public	无修饰	Private	Protect	final	abstract	static
类继承	可继承	只有同一包中的可继承	不能修饰类	不能修饰类	不能派生子类	一般可继承	不能修饰类
方法重载	可重载	可重载	不能重载	可重载	不可重载	可重载	可重载（修饰主函数就不能重载了）
成员变量（属性的隐藏）	父类属性被隐藏（使用super. 获取父类属性）	父类属性被隐藏（使用super. 获取父类属性）	子类不能直接访问父类的私有变量	父类属性被隐藏（使用super. 获取父类属性）	必须付初值	不能修饰成员变量	每个实例共享这个类变量

注： 无修饰的类中，其他包无法访问，所以他去包中的类无法继承。

1.2.2 实验目的

1. 掌握使用 JDK 开发 Java 程序的步骤。掌握 Java 程序的基本语法；熟练运用 java 的三个工具编译或运行程序。
2. 掌握 Java 程序面向对象编程的基本架构，会运用面向对象的思想编写 Java 程序。

3. 掌握 Java 中的继承机制及包（package）、接口（interface）等的设计方法

1.2.3 实验内容

内容一:使用 Eclipse 创建自己的聊天室项目,组织好项目结构.

内容二：编写一个学生和教师数据输入和显示程序，学生数据有编号、姓名、班号和成绩，教师数据有编号、姓名、职称和部门。要求将编号、姓名输入和显示设计成一个类 Person,并作为学生数据操作类 Student 和教师数据操作类 Teacher 的基类。

内容三：请设计 3 个类 ，分别是学生类 Student, 本科生类 Undergraduate, 研究生类 Postgraduate, 其中 Student 类是一个抽象类 ，它包含一些基本的学生信息如姓名、所学课程、课程成绩等 ，而 Undergraduate 类和 Postgraduate 都是 Student 类的子类 ，它们之间的主要差别是计算课程成绩等级的方法有所不同 ，研究生的标准要比本科生的标准高一些 ，如表 1-2 所示。

表 1 课程成绩等级

本科生标准		研究生标准	
80--100	优秀	90--100	优秀
70--80	良好	80--90	良好
60--70	一般	70--80	一般

50--60	及格	60--70	及格
50 以下	不及格	60 以下	不及格

假设某班级里既有本科生也有研究生，请编写程序统计出全班学生的成绩等级并显示出来。此题关键是设计一个学生数组，既能存放本科生对象，又能存放研究生对象。

1.2.5 实验思考

1. 面向对象的基本特征是什么?在你的作业中是如何体现的?
2. 什么是访问控制?为什么需要访问控制?
3. 思考接口和抽象类的区别.
4. 列出实现一个聊天室需要的类(从一个聊天室有哪些客观实物组成去考虑),并在下次上机之前实现这些实体类到自己项目中

1.2.6 实验总结

请结合 JAVA 程序设计课程中的相关理论，对已完成的实验课程所需理论进行总结。

要求：

实验地点与实验（软、硬件）环境

实验内容与原理

实验结果与结果分析

实验心得体会与建议（不少于 500 汉字）

实验二 Java IO 编程实验

实验十三 输入输出（一）

一、实验目的:

1. 理解 Java 输入输出的实现机制;
2. 掌握常用的输入输出类的功能及方法;
3. 掌握常用数据类型针对文件的输入输出方式。

二、实验内容：

1. double 型数组中保存有如下数据：1.23, 69.88, 34.45, 67.98, 2345.67。将数据以二进制形式保存到文件中。

```
public class Test {

    /**
     * @param args
     * @throws FileNotFoundException
     */

    public static void main(String[] args) throws Exception {

        // TODO 自动生成方法存根

        String f = "d:/a.txt";

        double q[] = {1.23,69.88,34.45,67.98,2345.67};


        FileOutputStream fos = new FileOutputStream(f);

        for(int i = 0;i<q.length;i++){

            fos.write((char)q[i]);
        }
    }
}
```



```

    }
}

}

```

2. 编程将以上数据以可打印形式保存到文件中。

```

import java.io.*;

public class Test {

    public static void main(String args[]) throws Exception{

        String n = "asdasdasdas";

        BufferedWriter bw = new BufferedWriter(new
FileWriter("d:/b.txt"));

        bw.write(n);

        bw.close();

    }

}

```

3. 读入 <http://www.sina.com.cn> 网页的内容，并以可读形式将其输出到一个文件中。

```

import java.net.URL;

import java.io.*;

public class Test {

    /**
     * @param args
     * @throws Exception
     */

    public static void main(String[] args) throws Exception {

        URL u = new URL("http://www.sina.com.cn");
    }
}

```

```
InputStream is = u.openStream();

is = new BufferedInputStream(is);


Reader r = new InputStreamReader(is);

FileOutputStream fos = new FileOutputStream("d:/c.html");

int c = 0;

while((c = r.read())!=-1){

    fos.write(c);

}

}
```

三、试验要求:

- 1、预习试验内容并写出上机报告。
- 2、实验中出现的问题及实验体会。

综合大作业：C/S 架构的聊天室设计与实现

5.1 完成网络聊天室应用

5.1.1 知识介绍

C/S 结构，即大家熟知的客户机和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通讯开销。目前大多数应用软件系统都是 Client/Server 形式的两层结构，由于现在的软件应用系统正在向分布式的 Web 应用发展，Web 和 Client/Server 应用都可以进行同样的业务处理，应用不同的模块共享逻辑组件；因此，内部的和外部的用户都可以访问新的和现有的应用系统，通过现有应用系统中的逻辑可以扩展出新的应用系统。这也就是目前应用系统的发展方向。

C/S 结构的基本原则是将计算机应用任务分解成多个子任务，由多台计算机分工完成，即采用“功能分布”原则。客户端完成数据处理，数据表示以及用户接口功能；服务器端完成 DBMS（数据库管理系统）的核心功能。这种客户请求服务、服务器提供服务的处理方式是一种新型的计算机应用模式。

Client 和 Server 常常分别处在相距很远的两台计算机上，Client 程序的任务是将用户的要求提交给 Server 程序，再将 Server 程序返回的结果以特定的形式显示给用户；Server 程序的任务是接收客户程序提出的服务请求，进行相应的处理，再将结果返回给客户程序。

首先必须强调的是 C/S 和 B/S 并没有本质的区别：B/S 是基于特定通信协议(HTTP)的 C/S 架构，也就是说 B/S 包含在 C/S 中，是特殊的 C/S 架构。

之所以在 C/S 架构上提出 B/S 架构，是为了满足瘦客户端、一体化客户端的需要，最终目的节约客户端更新、维护等的成本，及广域资源的共享。

5.1.2 实验目的

- 1、使用 Java 技术完成聊天室系统，深入学习使用 Java 语言。
- 2、使用 Java 的多线程机制，深入理解 Java 多线程技术的应用。
- 3、使用 GUI，对 JFrame 的深入学习。
- 4、使用网络编程，掌握基于 TCP 协议的 Socket 编程，了解 Socket 编程的协议约定，掌握简单应用协议的开发。
- 5、使用 C/S 架构，对网络编程有一定的了解。

5.1.3 实验内容

- 1、实现一个基于 socket 的简易 C/S 聊天室
- 2、给自己的聊天室加上 GUI

5.1.4 实验示例

1、服务器

```
import java.io.IOException;

import java.net.ServerSocket;
```

```

import java.net.Socket;

import java.util.ArrayList;

public class MainServer {

    public static ArrayList<Socket> socketList = new
ArrayList<Socket>();

    public static void main(String[] args) throws IOException {

        ServerSocket serverSocket = new ServerSocket(30000);

        while (true) {

            System.out.println("等待客户端的访问...");

            Socket socket = serverSocket.accept();

            socketList.add(socket);

            new Thread(new ServerThread(socket)).start();

        } } }

```

2、服务器线程

```

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

public class ServerThread implements Runnable {

    Socket socket = null;

    BufferedReader bufferedReader = null;

```

```

public ServerThread(Socket socket) throws IOException {

    this.socket = socket;

    bufferedReader      =      new      BufferedReader(new
InputStreamReader(socket.getInputStream()));

}

@Override

public void run() {

    String content = null;

    try {

        while((content = readFromClient()) != null){

            for (Socket socket:MainServer.socketList) {

                PrintStream      printStream      =      new
PrintStream(socket.getOutputStream());

                printStream.println("来自客户端: "+content);

            }

        }

    } catch (IOException e) {

        e.printStackTrace();

    }

}

```

```

private String readFromClient(){
    try {
        return bufferedReader.readLine();
    } catch (IOException e) {
        MainServer.socketList.remove(socket);
    }
    return null;
}
}

```

3、客户端入口：

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.Socket;
import java.net.UnknownHostException;

public class MainClient {

    public static void main(String[] args) throws
    UnknownHostException, IOException {

        Socket clientSocket = new Socket("127.0.0.1", 30000);

```

```

        PrintStream      printStream      =      new
PrintStream(clientSocket.getOutputStream());

        String line = null;

        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));

        while ((line = bufferedReader.readLine()) != null) {

            printStream.println(line);

        }

    }
}

```

4、客户端线程

```

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class ClientThread implements Runnable {

    private Socket socket;

    BufferedReader bufferedReader = null;

```



```

public ClientThread(Socket socket) throws IOException{

    this.socket = socket;

    bufferedReader      =      new      BufferedReader(new
InputStreamReader(this.socket.getInputStream()));

}

@Override

public void run() {

    String conString = null;

    try {

        while ((conString = bufferedReader.readLine()) != null) {

            System.out.println(" 来 自 服 务 器 的 广 播  :

"+conString);

        }

    } catch (IOException e) {

        e.printStackTrace();

    }

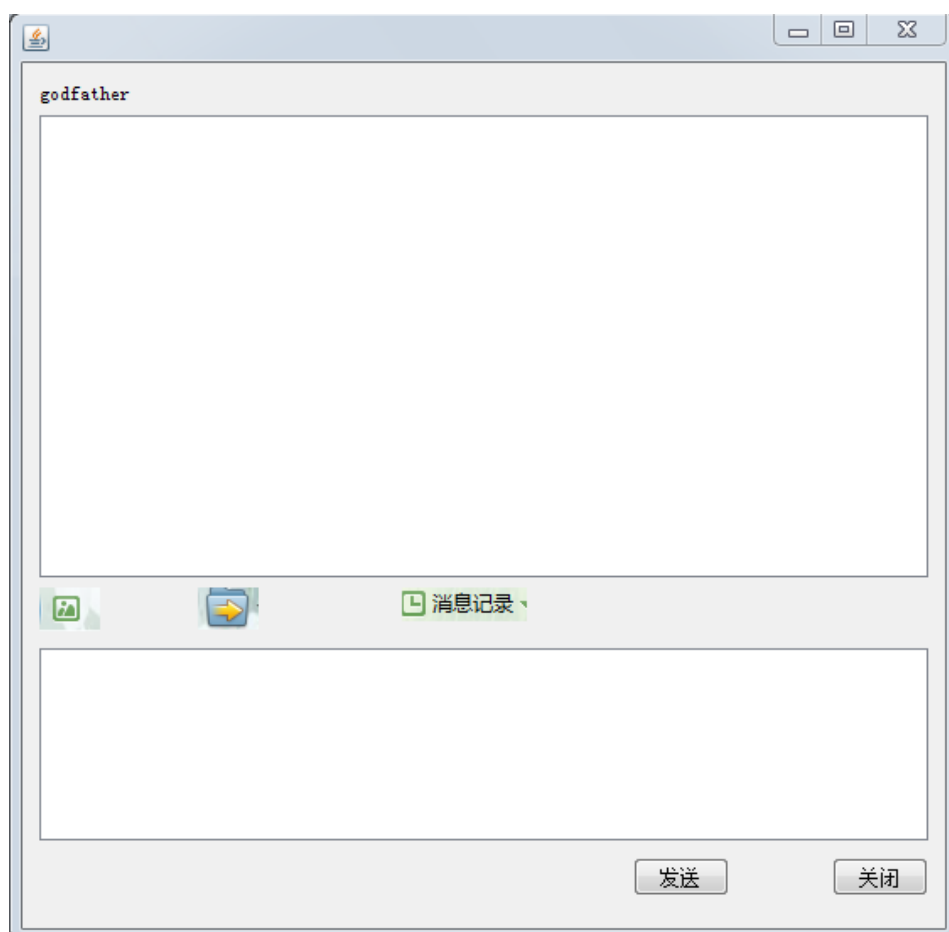
}

}

```

GUI 实现:





5.1.5 思考

- 1.加入聊天记录储存数据库
- 2.实现私聊.

5.1.6 实验总结

采用 C/S 架构设计的聊天室，使用增量迭代方式开发，通过本次实验，对 C/S 架构有了较深的认识，也对多线程编程、网络编程、GUI 的认识更清晰，能够将所有学过的 Java 知识综合运用，完成应用开发任务。