

**Southwest university of science and technology**

本科毕业设计（论文）

排序算法网络演示软件设计与实现

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术 |
| 专业名称 | 计算机科学与技术 |
| 学生姓名 |  |
| 学号 |  |
| 指导教师 | 孙敏 讲师 |

二〇一五年六月

排序算法网络演示软件设计与实现

摘要： 排序在人们的日常生活和学习、科研、生产等各个方面有着重要的应用。排序算法，就是如何使得记录按照要求排列的方法。排序算法在很多领域得到相当地重视，尤其是在大量数据的处理方面。一个优秀的算法可以节省大量的资源。在各个领域中考虑到数据的各种限制和规范，要得到一个符合实际的优秀算法，得经过大量的推理和分析。因此掌握常用的排序算法是很必要的。本软件采用java2SE为开发工具，实现四种不同排序算法即：快速排序、冒泡排序、直接插入排序、直接选择排序、的排序算法的动画演示。软件界面美观，操作简单，可作为排序可视化教学演示软件。

关键词：排序演示；排序；java2SE

**Sorting algorithm network demo software design and implementation**

**Abstract:**Order in people's daily lives and learning, research, production and other aspects of important applications. Sorting algorithm is how to make the recording method in accordance with the requirements of arrangement. Sorting algorithms in many fields quite seriously, especially in dealing with large amounts of data. A good algorithm can save a lot of resources. Consider in various fields to a variety of restrictions and specifications data, to get a realistic outstanding algorithm, it was after a lot of reasoning and analysis. So have the sorting algorithm used is very necessary. Java2SE for the system development tools, to achieve eight different sorting algorithm that is: Quick Sort, Bubble Sort, Heap Sort, Insertion Sort directly Hill to sort, direct selection sort, merge sort to sort the order of the base presentation.

**Key words**：Algorithm demo,Sort,Java2SE

目录

[第1章 引言 1](#_Toc11549)

[1.1选题背景与意义 1](#_Toc18097)

[1.2国内外研究现状 1](#_Toc16817)

[1.3本课题的研究内容 1](#_Toc8299)

[1.4开发工具介绍 2](#_Toc24641)

[第2章 排序算法和线程 3](#_Toc27423)

[2.1排序算法 3](#_Toc4113)

[2.1.1排序算法的基本概念 3](#_Toc8841)

[2.1.2排序算法简介 4](#_Toc12649)

[2.1.3 排序算法的比较 10](#_Toc4956)

[2.2线程 10](#_Toc23932)

[2.2.1线程的基本概念 10](#_Toc7208)

[2.2.2 线程的特点 11](#_Toc556)

[第3章 软件的分析和设计 13](#_Toc15986)

[3.1 设计目标 13](#_Toc349)

[3.2需求概述 13](#_Toc18772)

[3.2.1 功能性需求 13](#_Toc6008)

[3.2.2 非功能性需求 16](#_Toc29286)

[3.3 软件总体设计 17](#_Toc11069)

[3.3.1 软件总体概述 17](#_Toc5831)

[3.3.2优化分析模型 18](#_Toc31016)

[第4章 软件的详细设计 20](#_Toc28021)

[4.1软件范围 20](#_Toc24303)

[4.2 用例分析 20](#_Toc32084)

[4.2.1识别参与者 20](#_Toc28633)

[4.2.2识别用例 20](#_Toc31901)

[4.2.3构建用例图 21](#_Toc266)

[4.2.4细化用例图 21](#_Toc11398)

[4.3 构建类模型 22](#_Toc5369)

[4.3.1确定类 22](#_Toc7007)

[4.3.2确定属性 23](#_Toc1949)

[4.4 状态模型 24](#_Toc18564)

[4.4.1确定状态 24](#_Toc10528)

[4.4.2构建事件跟踪图 25](#_Toc16580)

[4.4.3构建状态图 26](#_Toc8989)

[4.5服务分析 27](#_Toc16183)

[4.5.1服务分析 27](#_Toc9380)

[4.5.2软件最终类 28](#_Toc32158)

[4.6 类的设计 28](#_Toc29602)

[4.6.2类之间的联系 31](#_Toc30895)

[4.6.3软件运行流程图 33](#_Toc13928)

[4.7模块的详细设计 34](#_Toc20496)

[4.7.1输入模块 34](#_Toc7763)

[4.7.2选择模块 35](#_Toc26307)

[4.7.3处理模块 36](#_Toc9460)

[4.7.4显示模块 39](#_Toc20560)

[第5章 软件实现和测试 42](#_Toc22303)

[5.1软件实现 42](#_Toc14328)

[5.1.1软件设计 42](#_Toc17867)

[5.2软件测试 43](#_Toc6828)

[5.2.1测试环境 43](#_Toc12975)

[5.2.2用户界面测试 43](#_Toc12709)

[5.2.3简单功能测试 44](#_Toc32504)

[结论和展望 46](#_Toc6859)

[完成的任务 46](#_Toc21656)

[收获总结 46](#_Toc30036)

[展望未来 47](#_Toc1771)

[致谢 48](#_Toc7222)

[参考文献 49](#_Toc19924)

# 第1章 引言

## 1.1选题背景与意义

由于排序在计算机图形、计算机辅助设计、机器人、模式识别、基因排序工程及统计学等领域具有广泛应用，所以对排序的研究既有理论上的重要意义，又有实际应用价值。再加上现在信息产业的迅速发展，信息的流通量越来越大，如此庞大并且杂乱无章的信息数据十分难以管理和查询，就更加需要一种十分快捷而有效的编排手段来整理这些数据信息，让工作效率得以提高。

本课题主要用Java的多线程技术实现的操作软件辅助教学软件，以动画形式表示算法的执行过程,并且还可以同步显示算法的源代码以及其中变量的变化,使算法的执行过程一目了然，具有很好的教学效果。该教学软件使数据结构的抽象的概念、复杂的算法、操作实现过程用图形及动画展现出来，使教与学都变得更加容易，也提出了计算机软件课程实验教学的一种新方法，对学生深刻理解算法执行过程有很大帮助。

## 1.2国内外研究现状

随着计算机技术的日益发展，其应用早已不局限于简单的数值运算。而设计到问题的分析、数据结构框架的设计以及插入、删除、排序、查找等复杂的非数值处理和操作。排序算法的学习就是为以后利用计算机资源高效地开发非数值处理的计算机程序打下坚实的理论、方法和技术基础。近来国内外专家学者们对排序算法又有了新的研究和发现。例如：山东大学的张峰和张金屯两位教授共同研究了《我国植被数量分类和排序研究进展》这一课题，很值得有关部门去学习和研究。

## 1.3本课题的研究内容

本文主要是介绍一下几种排序算法的基本概念，然后利用java来开发一个数据结构的动画演示软件。该软件主要是通过操作数据结构中的数据把常见的排序算法（冒泡排序、选择排序、直接插入排序、快速排序）表示出来。

## 1.4开发工具介绍

(1) Java语言

Java是一种可以撰写跨[平台](http://baike.baidu.com/view/58664.htm)[应用](http://baike.baidu.com/view/220910.htm)[软件](http://baike.baidu.com/subview/37/6030295.htm)的面向对象的[程序](http://baike.baidu.com/subview/17674/17674.htm)[设计](http://baike.baidu.com/view/14417.htm)[语言](http://baike.baidu.com/subview/9793/10663680.htm)，是由[Sun Microsystems](http://baike.baidu.com/view/1061350.htm)公司于1995年5月推出的Java程序设计语言和[Java平台](http://baike.baidu.com/view/209634.htm)（即[JavaEE](http://baike.baidu.com/view/169790.htm), [JavaME](http://baike.baidu.com/view/1560387.htm), JavaSE）的总称。Java自面世后就非常流行，发展迅速，对[C++](http://baike.baidu.com/view/824.htm)语言形成了有力冲击。Java技术具有卓越的通用性、高效性、平台移植性和[安全](http://baike.baidu.com/view/4547.htm)性，广泛应用于个人[PC](http://baike.baidu.com/subview/13621/4882082.htm)、[数据中心](http://baike.baidu.com/view/754520.htm)、[游戏](http://baike.baidu.com/subview/2468/8069356.htm)控制台、[科学](http://baike.baidu.com/view/3805.htm)[超级计算机](http://baike.baidu.com/view/296324.htm)、[移动电话](http://baike.baidu.com/view/49217.htm)和[互联网](http://baike.baidu.com/view/6825.htm)，同时拥有全球最大的开发者专业社群。在全球[云计算](http://baike.baidu.com/view/1316082.htm)和[移动互联网](http://baike.baidu.com/view/1168245.htm)的产业环境下，[Java](http://baike.baidu.com/subview/29/12654100.htm)更具备了显著优势和广阔前景。[Java](http://baike.baidu.com/subview/29/12654100.htm)分为三个体系JavaSE（[J2SE](http://baike.baidu.com/view/160728.htm)）（Java2 Platform Standard Edition，java平台标准版），JavaEE([J2EE](http://baike.baidu.com/view/1507.htm))(Java 2 Platform,Enterprise Edition，java平台[企业](http://baike.baidu.com/view/38340.htm)版)，JavaME([J2ME](http://baike.baidu.com/view/7125.htm))(Java 2 Platform Micro Edition，java平台微型版)。

(2)Eclipe

Eclipse是著名的跨平台的自由集成开发环境（IDE），最初主要用来Java语言开发，但是目前亦有人通过插件使其作为其他计算机语言比如C++和Python的开发工具。Eclipse原本只是一个框架平台，提供插件开发环境（Plug-in Development Environment，PDE），这个组件主要针对希望扩展 Eclipse 的软件开发人员，允许构建与 Eclipse 环境无缝集成的工具，众多插件的支持使得Eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。由于 Eclipse 中的每样东西都是插件，给 Eclipse 提供插件，为用户提供一致和统一的集成开发环境，使所有工具开发人员都具有同等的发挥场所。

# 第2章 排序算法和线程

## 2.1排序算法

### 2.1.1排序算法的基本概念

2.1.1.1排序

所谓[排序](http://baike.baidu.com/view/58783.htm" \t "_blank)，就是使一串记录，按照其中的某个或某些关键字的大小，递增或递减的排列起来的操作。排序算法，就是如何使得记录按照要求排列的方法。排序算法在很多领域得到相当地重视，尤其是在大量数据的处理方面。一个优秀的算法可以节省大量的资源。在各个领域中考虑到数据的各种限制和规范，要得到一个符合实际的优秀算法，得经过大量的推理和分析。

2.1.1.2排序分类

根据排序元素索在位置的不同，排序分为内排位和外排序

内排序是指在排序过程中，所有元素调到内存中进行的排序，成为内排序。内排序是排序的接触。内排序效率用比较次数来衡量。按所用策略不同，内排序又可分为插入排序、选择排序、交换排序、归并排序及基数排序等几大类。

外排序是指排序过程中还需访问外存储器，足够大的元素序列，因不能完全放入内存，只能使用外排序。

2.1.1.3 排序的性能

1. 执行时间和所需的辅助时间

若排序算法所需的辅助空间并不依赖于问题的规模n，即辅助空间是o(1),则称之为就地排序（In-place Sort）。非就地排序一般要求的辅助空间为o(n).

1. 算法本身的复杂成都
2. 排序算法的时间开销：大多数排序算法的时间开销主要是关键字之间的比较和记录的移动，有的排序算法其执行时间不仅依赖于问题的规模，还取决于输入实力中数据的状态。

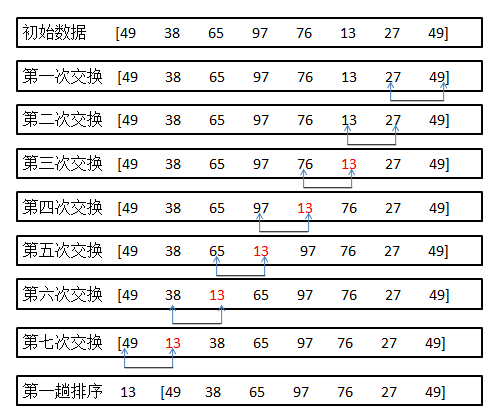
### 2.1.2排序算法简介

2.1.2.1 冒泡排序

1. 基本原理

冒泡排序算法的运作如下：（从后往前）

1. 比较相邻的元素。如果第一个比第二个大，就交换他们两个。
2. 对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。在这一点，最后的元素应该会是最大的数。
3. 针对所有的元素重复以上的步骤，除了最后一个。
4. 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。



**图2- 1 冒泡排序算法的过程**

1. 算法描述

#public void sort(int[] data) {

# for (int j = 0; j < data.length; j++) {

# for (int i = data.length - 1; i >= j + 1; i--) {

# if (data[i - 1] > data[i]) {

# data[i - 1] = data[i - 1] + data[i];

# data[i] = data[i - 1] - data[i];

# data[i - 1] -= data[i];

# }

# }

# }

#}

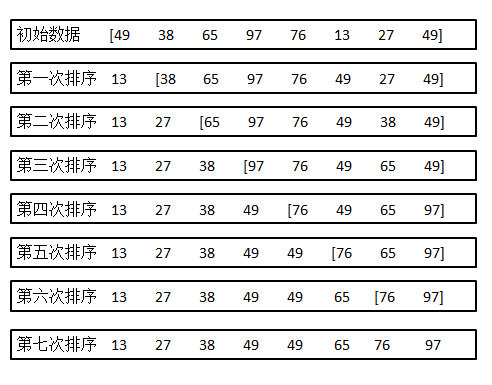
1. 时间复杂度分析

若文件的初始状态是正序的，一趟扫描即可完成排序。所需的关键字比较次数C 和记录移动次数M均达到最小值：Cmin = n-1，Mmin = 0。 所以，冒泡排序最好的[时间复杂度](http://baike.baidu.com/view/104946.htm" \t "_blank)为O(n) 。若初始文件是反序的，需要进行n-1 趟排序。每趟排序要进行n-i次关键字的比较(1≤i≤n-1)，且每次比较都必须移动记录三次来达到交换记录位置。在这种情况下，比较和移动次数均达到最大值：Cmax = n(n-1)/2 = O(n^2),Mmax = 3n(n-1)/2 = O(n^2)。冒泡排序的最坏时间复杂度为O(n^2) 。综上，因此冒泡排序总的平均时间复杂度为O(n^2)。

2.1.2.2 选择排序

1. 基本原理

每一次从待排序的[数据元素](http://baike.baidu.com/view/38785.htm" \t "_blank)中选出最小（或最大）的一个元素，存放在序列的起始位置，直到全部待排序的数据元素排完。



**图2- 2 选择排序算法的过程**

1. 算法描述

#public void selectSort(int[] data) {

# for (int j = 0; j <= data.length - 2; j++) {

# int lowestIndex = j;

# for (int i = data.length - 1; i > j; i--) {

# if (data[i].compareTo(data[lowestIndex]) < 0) {

# lowestIndex = i;

# }

# }

# int temp = data[j];

# data[j] = data[lowestIndex];

# data[lowestIndex] = temp;

# }

#}

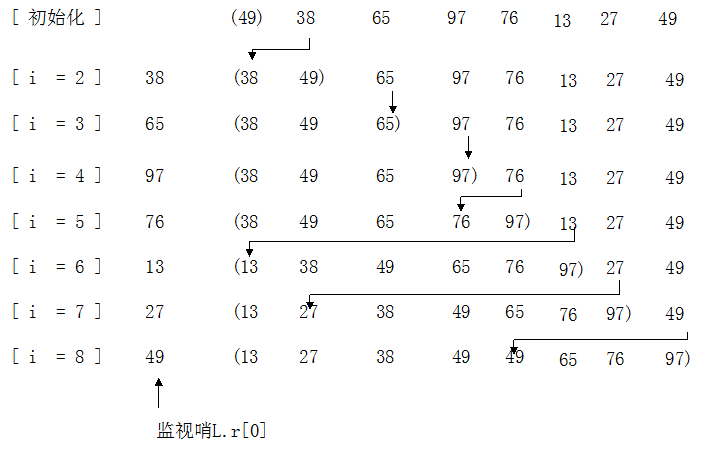
1. 时间复杂度分析

该算法运行时间与元素的初始排列无关。不论初始排列如何，该算法都必须执行n-1趟，每趟执行n-i-1次关键字的比较，这样总的比较次数为：所以，简单选择排序的最好、最坏和平均情况的时间复杂度都为O(n2)。

2.1.2.3 插入排序

1. 基本原理

假设待排序的n个记录{L0，L1，…，Ln}顺序存放在数组中，直接插入法在插入记录Li(i=1，2，…，n-1)时，记录被划分为两个区间[L0,Li-1]和[Li+1,Ln-1],其中，前一个子区间已经排好序，后一个子区间是当前未排序的部分，将关键码Ki与Ki-1Ki-2，…，K0依次比较，找出应该插入的位置，将记录Li插，然后将剩下的i-1个元素按关键词大小依次插入该有序序列，没插入一个元素后依然保持该序列有序，经过i-1趟排序后即成为有序序列。每次将一个待排序的记录，按其关键字大小插入到前面已经排好序的子文件中的适当位置，直到全部记录插入完成为止，为了在查找插入位置的过程中避免数组下标出界，在l[0]处设置监视哨。



**图2- 3 插入排序算法的过程**

1. 算法描述

#public void inertSort(int[] data) {

# for (int j = 1; j < data.length; j++) {

# int i = j - 1;

# int key = data[j];

# while (i >= 0 && data[i] > key) {

# data[i + 1] = data[i];

# data[i] = key;

# i -= 1;

# }

# }

#}

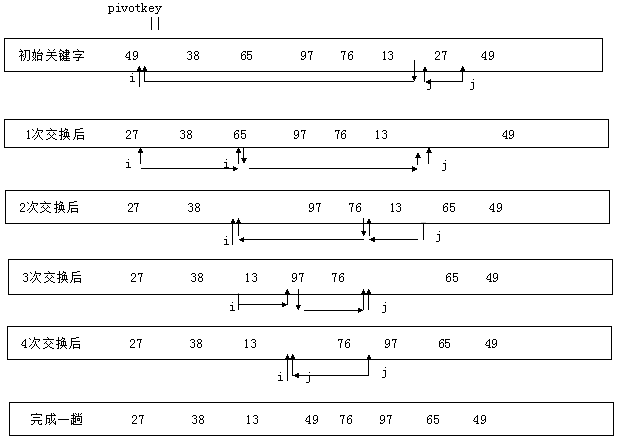
1. 时间复杂度分析

直接插入排序算法必须进行n-1趟。最好情况下，即初始序列有序，执行n-1趟，但每一趟只比较一次，移动元素两次，总的比较次数是(n-1)，移动元素次数是2(n-1)。因此最好情况下的时间复杂度就是O(n)。最坏情况(非递增)下，最多比较i次，因此需要的比较次数是：所以，时间复杂度为O(n2)。

2.1.2.4 快速排序

1. 基本原理

对起泡排序的一种改进。它的基本思想是，通过一趟排序将待排记录分割成独立的两部分，其中一部分记录的关键字均比另一部分记录的关键字小则可分别对两部分记录继续进行排序，以达到整个序列有序。假设待排序的序列为{ K.r[s],K.r[s+1],...,K.r[t]},首先任意选取一个记录（通常选第一个）作为支点，然后按下述原则重新排列其余记录：将所有关键字比它小的记录都安置在它的位置之前，将所有关键字较它打的记录都安置在它的位置之后。由此可以将位置i作分界线，将序列{ K.r[s],K.r[s+1],...,K.r[t]}分割成两个子序列{ K.r[s],K.r[s+1],...,K.r[i-1]}和{ K.r[i+1],K.r[s+1],...,K.r[t]}。这个过程称一趟快速排序。



**图2- 4快速排序算法的一趟排序过程**

1. 算法描述

#void quick\_sort(int data[], int l, int r)

#{

# if (l < r)

# {

# int i = l, j = r, x = data[l];

# while (i < j)

# {

# while(i < j && data[j] >= x)

# j--;

# if(i < j)

# data[i++] = data[j];

# while(i < j && data[i] < x)

# i++;

# if(i < j)

# data[j--] = data[i];

# }

# data[i] = x;

# quick\_sort(data, l, i - 1);

# quick\_sort(data, i + 1, r);

# }

#}

1. 时间复杂度分析

如果每一次分划操作后，左、右两个子序列的长度基本相等，则快速排序的效率最高，其最好情况时间复杂度为O(nlog2n)；反之，如果每次分划操作所产生的两个子序列，其中之一为空序列，此时，快速排序效率最低，其最坏情况时间复杂度为O(n2)。如果选择左边第一个元素为主元，则快速排序的最坏情况发生在原始序列正向有序或反向有序时。快速排序的平均情况时间复杂度为O(nlog2n)。

### 2.1.3 排序算法的比较

在解决实际问题的时候，采用的排序算法有很多种，而且这些排序算法各有优缺点，难以得出哪个算法最好，哪个算法最坏的结论。因此排序算法的选用要视具体场合而定。下面我们给出一个排序算法的特点表



**图2- 5四种排序算法的比较**

## 2.2线程

### 2.2.1线程的基本概念

线程，有时被称为轻量级进程(Lightweight Process，LWP），是程序执行流的最小单元。一个标准的线程由线程ID，当前指令[指针](http://baike.baidu.com/view/159417.htm" \t "http://baike.baidu.com/_blank)(PC），[寄存器](http://baike.baidu.com/view/6159.htm" \t "http://baike.baidu.com/_blank)集合和[堆栈](http://baike.baidu.com/view/93201.htm" \t "http://baike.baidu.com/_blank)组成。另外，线程是进程中的一个实体，是被系统独立调度和分派的基本单位，线程自己不拥有系统资源，只拥有一点儿在运行中必不可少的资源，但它可与同属一个进程的其它线程共享进程所拥有的全部资源。一个线程可以创建和撤消另一个线程，同一进程中的多个线程之间可以并发执行。由于线程之间的相互制约，致使线程在运行中呈现出间断性。线程也有[就绪](http://baike.baidu.com/view/654230.htm" \t "http://baike.baidu.com/_blank)、[阻塞](http://baike.baidu.com/view/497285.htm" \t "http://baike.baidu.com/_blank)和[运行](http://baike.baidu.com/view/1026025.htm" \t "http://baike.baidu.com/_blank)三种基本状态。就绪状态是指线程具备运行的所有条件，逻辑上可以运行，在等待处理机；运行状态是指线程占有处理机正在运行；阻塞状态是指线程在等待一个事件（如某个信号量），逻辑上不可执行。每一个程序都至少有一个线程，若程序只有一个线程，那就是程序本身。

线程是程序中一个单一的顺序控制流程。进程内一个相对独立的、可调度的执行单元，是系统独立调度和分派CPU的基本单位指[运行](http://baike.baidu.com/view/1026025.htm" \t "http://baike.baidu.com/_blank)中的程序的调度单位。在单个程序中同时运行多个线程完成不同的工作，称为[多线程](http://baike.baidu.com/view/65706.htm" \t "http://baike.baidu.com/_blank)。

### 2.2.2 线程的特点

在[多线程](http://baike.baidu.com/view/65706.htm" \t "http://baike.baidu.com/_blank)OS中，通常是在一个[进程](http://baike.baidu.com/view/19746.htm" \t "http://baike.baidu.com/_blank)中包括多个线程，每个线程都是作为利用CPU的基本单位，是花费最小开销的实体。线程具有以下属性。

1）轻型实体

线程中的实体基本上不拥有[系统资源](http://baike.baidu.com/view/53557.htm" \t "http://baike.baidu.com/_blank)，只是有一点必不可少的、能保证独立[运行](http://baike.baidu.com/view/1026025.htm" \t "http://baike.baidu.com/_blank)的资源。

线程的实体包括程序、数据和TCB。线程是动态概念，它的动态特性由线程控制块TCB（Thread Control Block）描述。TCB包括以下信息：

（1）线程状态。

（2）当线程不运行时，被保存的现场资源。

（3）一组执行堆栈。

（4）存放每个线程的局部变量主存区。

（5）访问同一个进程中的主存和其它资源。

用于指示被执行指令序列的[程序计数器](http://baike.baidu.com/view/178145.htm" \t "http://baike.baidu.com/_blank)、保留[局部变量](http://baike.baidu.com/view/552847.htm" \t "http://baike.baidu.com/_blank)、少数状态参数和[返回地址](http://baike.baidu.com/view/11774464.htm" \t "http://baike.baidu.com/_blank)等的一组[寄存器](http://baike.baidu.com/view/6159.htm" \t "http://baike.baidu.com/_blank)和[堆栈](http://baike.baidu.com/view/93201.htm" \t "http://baike.baidu.com/_blank)。

2）独立调度和分派的基本单位。

在多线程OS中，线程是能独立[运行](http://baike.baidu.com/view/1026025.htm" \t "http://baike.baidu.com/_blank)的基本单位，因而也是独立调度和分派的基本单位。由于线程很“轻”，故线程的切换非常迅速且开销小（在同一[进程](http://baike.baidu.com/view/19746.htm" \t "http://baike.baidu.com/_blank)中的）。

3）可并发执行。

在一个[进程](http://baike.baidu.com/view/19746.htm" \t "http://baike.baidu.com/_blank)中的多个线程之间，可以并发执行，甚至允许在一个进程中所有线程都能并发执行；同样，不同进程中的线程也能并发执行，充分利用和发挥了[处理机](http://baike.baidu.com/view/2107226.htm" \t "http://baike.baidu.com/_blank)与外围设备并行工作的能力。

1. 共享[进程](http://baike.baidu.com/view/19746.htm" \t "http://baike.baidu.com/_blank)资源。

在同一[进程](http://baike.baidu.com/view/19746.htm" \t "http://baike.baidu.com/_blank)中的各个线程，都可以共享该进程所拥有的资源，这首先表现在：所有线程都具有相同的[地址空间](http://baike.baidu.com/view/1507129.htm" \t "http://baike.baidu.com/_blank)（进程的地址空间），这意味着，线程可以访问该地址空间的每一个虚地址；此外，还可以访问进程所拥有的已打开[文件](http://baike.baidu.com/view/345685.htm" \t "http://baike.baidu.com/_blank)、定时器、[信号量](http://baike.baidu.com/view/1303265.htm" \t "http://baike.baidu.com/_blank)机构等。由于同一个[进程](http://baike.baidu.com/view/19746.htm" \t "http://baike.baidu.com/_blank)内的线程[共享内存](http://baike.baidu.com/view/120892.htm" \t "http://baike.baidu.com/_blank)和[文件](http://baike.baidu.com/view/345685.htm" \t "http://baike.baidu.com/_blank)，所以线程之间互相通信不必调用[内核](http://baike.baidu.com/view/1366.htm" \t "http://baike.baidu.com/_blank)。

# 第3章 软件的分析和设计

## 3.1 设计目标

本软件是在Windows平台下，使用java开发排序算法的动画演示。用户通过软件可以更加深刻理解算法执行过程，主要包含的内容有一下几方面：

1. 实现用户进入该界面有可选择按钮，分为四种情况：插入排序、冒泡排序、选择排序、快速排序。
2. 用户选择之后，通过用绘图的方式实现该算法的动态演示
3. 给出代码区,程序执行步骤和演示步骤能由代码区反映
4. 显示关键变量的值变化情况

## 3.2需求概述

### 3.2.1 功能性需求

1. 选择排序算法功能

**表3- 1选择排序算法功能**

|  |  |
| --- | --- |
| 名称 | 选择排序算法功能 |
| 功能描述 | 用户通过按钮就自主的选择一种排序算法进行演示 |
| 界面描述 | 软件主页面有一个选择按钮 |
| 输入项目 | 一种排序算法的名称 |
| 输出项目 | 一种排序算法的名称 |
| 备注 | 输入和输出的算法是一一对应的 |

1. 绘图区动画演示功能

**表3- 2绘图区动画演示功能**

|  |  |
| --- | --- |
| 名称 | 绘图区动画演示功能 |
| 功能描述 | 通过用户选择的排序算法，能准确的演示数据排序运行的全过程 |
| 界面描述 | 1、软件主页面划分出来绘图区域  2、里面包含有所需数据的方块。 |
| 输入项目 | 一种排序算法的名称 |
| 输出项目 | 一种排序算法的动画演示 |
| 备注 | 输入和输出的算法是一一对应的 |

1. 代码区同步代码功能

**表3- 3代码区同步代码功能**

|  |  |
| --- | --- |
| 名称 | 代码区同步代码功能 |
| 功能描述 | 通过排序算法运行步骤，来同步显示代码运行到的行数 |
| 界面描述 | 1、软件主页面划分出来文本区域  2、里面含有正在演示的代码 |
| 输入项目 | 代码运行的步骤 |
| 输出项目 | 同步选择到代码相应的行数 |
| 备注 |  |

1. 变量区同步变量功能

**表3- 4变量区同步变量功能**

|  |  |
| --- | --- |
| 名称 | 变量区同步变量功能 |
| 功能描述 | 通过排序算法运行步骤，来同步显示各个变量值的情况 |
| 界面描述 | 1、 软件主页面划分出来变量区域  2、里面含有显示的代码变量值 |
| 输入项目 | 代码运行的步骤 |
| 输出项目 | 同步代码变量值 |
| 备注 |  |

1. 处理数据功能

**表3- 5处理数据功能**

|  |  |
| --- | --- |
| 名称 | 处理数据功能 |
| 功能描述 | 进行数据的初始化以及处理等。 |
| 界面描述 | 无 |
| 输入项目 | 数据的数量 |
| 输出项目 | 同样数量的数据，并初始化完毕。 |
| 备注 |  |

### 3.2.2 非功能性需求

1. 时间特性要求

由于数据的的随机性，另外还由于排序算法的差异性，导致排序算法的的时间差异比较大，所以排序所完成的时间是不确定的。

1. 软件环境要求

**表3- 6软件环境要求**

|  |  |
| --- | --- |
| 软件环境 | WIN7 32/64 |
| 设计平台 | Eclipse |
| 接口 | Jdk6.0及以上的开发包 |

以上配置仅为开发者开发和测试过程中的实际开发和运行环境，我们保证在预装了上述版本（或更新版本）软件的计算机中能够运行本软件。

1. 硬件环境要求

**表3- 7硬件环境要求**

|  |  |
| --- | --- |
| 处理器 | 550MHz Pentium/Celeron系列，AMD K6/Athlon/Duron系列或性能优于上述系列的其它处理器 |
| 内存 | 512MB及以上 |
| 其它设备 | 视具体情况需要 |

以上配置仅为开发者建议能够运行本软件的最低配置，由于实际环境中不可预见的因素影响，我们不保证在上述配置的计算机上均能正常运行，但开发过程中，项目组成员需要充分考虑环境因素可能对本软件正常运行的影响，并对软件兼容性做出必要的改进，以满足大多数运行环境。

1. 输入输出要求

软件要求输入的是用户的操作请求和排序算法信息，输出的是排序算法的动画演示、同步代码以及变量值。

1. 界面要求

软件要求显示用户的基本信息和提供用户操作的接口，所以操作界面要求简洁美观，功能明显和易操作。

## 3.3 软件总体设计

### 3.3.1 软件总体概述

本软件主要包含用户选择排序算法模块，排序算法动画演示模块，排序算法代码同步模块和变量同步模块，数据处理模块。用户通过选择一个排序算法，然后后台进行数据的初始化，然后运行到每一步的同时，还要进行绘画以及代码的同步，同时还要进行数值同步变化。只需要用户进行算法的选择，一切自动的进行。软件整体的层次如下图所示



**图3- 1软件整体的层次图**

1. 排序算法演示处理

排序算法演示模块能够根据用户选择的排序算法对数据进行排序，动态的显示出排序过程。

1. 代码显示

进行排序算法的同时，也 显示用户选择的排序算法的代码步骤。

1. 数值显示

进行排序算法的同时，也显示用户选择的排序算法的数值表，同步数值的更新。

1. 数据处理

进行数据初始化。

### 3.3.2优化分析模型

本软件的功能就是将用户输入的数据在用户选择排序算法过程经过一系列的处理最后将得到的排序演示和排序算法的代码显示给用户。所以可以将本软件划分为四个模块。输入模块、选择模块、处理模块和显示模块。输入模块主要是实现了待排序数据的长度输入、选择模块主要是实现了排序算法的选择、处理模块主要是处理数据排序等，显示模块主要是把最后得到的排序演示和排序算法的代码显示在界面上。



**图3- 2 软件的模块图**

输入模块主要是当用户输入数据时接受数据；选择模块主要是当用户选择排序算法时接受选择；处理模块主要是排序算法的代码显示和绘制动画以及数值表的显示；显示模块主要是将排序算法的动画演示和代码同步显示以及数值的同步显示。

如下图给几个模块进行了详细结构



**图**3- 3 **软件细化模块图**

# 第4章 软件的详细设计

## 4.1软件范围

对于排序算法演示，本软件用用户输入的数据进行各种排序，并显示各排序算法的代码。

## 4.2 用例分析

### 4.2.1识别参与者

参与者就是软件在执行的过程中不可缺少的且与软件有交互行为的外部实体、对象或软件。

1. 用户打开软件界面进行想要的操作或不进行操作。
2. 用户输入数据。
3. 用户可以从界面上选择所要进行排序演示。
4. 软件将数据的排序演示过程和用户所选的排序算法的代码呈现给用户。
5. 软件推出

经过上述分析，发现软件在执行的过程中只有两个参与者，它们就是用户和界面。



**图4- 1软件参与者**

### 4.2.2识别用例

用例是软件执行的外部可见的产生对参与者有价值结果的动作序列，基于问题陈述给出以下分析并从中发现软件用例。

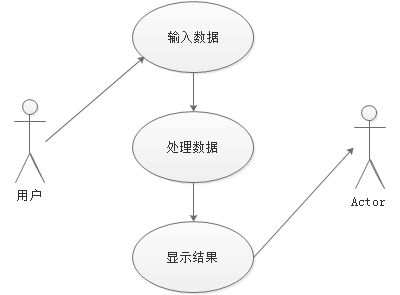
1. 使用者打开软件界面，输入数据并从中选择排序算法。
2. 软件显示相应排序演示操作并显示相应的排序算法。、

从上述分析中我们可以初步识别出软件 用例：输入数据、选择排序算法、排序演示、显示代码。



**图4- 2软件用例**

### 4.2.3构建用例图



**图4- 3软件用例图**

### 4.2.4细化用例图

将用例细化可得到分解的用例，将“排序演示”用例分解为：

1. 选择排序算法

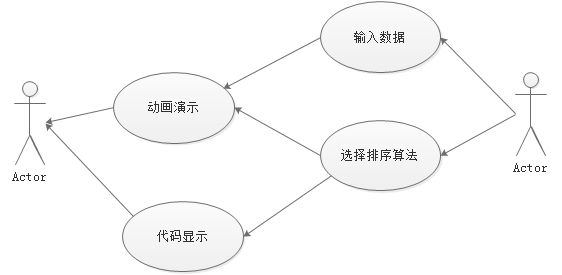
从界面上选择要进行排序的排序算法

1. 确定数据的比较顺序

根据选择的排序算法确定数据比较顺序。

1. 演示排序过程

显示选择排序算法对输入数据的排序过程和排序算法代码



**图4- 4细化后的用例图**

## 4.3 构建类模型

### 4.3.1确定类

4.3.1.1寻找类

根据名词识别法从需求陈述中获取如下类与对象：软件、界面、输入数据、数据、排序代码、排序演示 算法演示过程、代码、数据链。

4.3.1.2 筛选类

1. 无关类或对象的删除

本软件进行特征提取只关心数据、算法、排序代码、算法演示过程等处理过程，其他的都不是本软件关心的。在上面获取的对象中 “软件”和“用户”和本软件内部处理无关，所以需要删除。

1. 与实现无关删除
2. 重复删除

排序代码和代码在此都表示排序算法代码，所以删除排序代码。

1. 最终确定的类

经过以上几步的筛选且由于功能简单的确定为四个类：数据，绘图，数据链，界面。

4.3.1.3准备数据字典

上一步确定了四个类，对这两个类进行介绍：

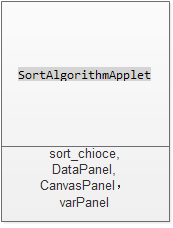
* + - 1. 数据：存储数据的各个属性
      2. 数据链：数据是数据链中的一个单位，主要是存储产生的随机数。
      3. 绘图： 绘制动画演示的一个工具、
      4. 界面：主要是呈现给用户的图像界面，把数据排序的演示过程和排序算法的代码呈现给用户。

### 4.3.2确定属性

属性是对象或类的数值特性及对象或类的性质，借助属性能够对对象和类的结构有更清晰的认识，以下来分析本软件所属类的属性。

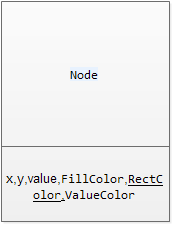
本软件有四个类：数据，数据链，绘图，界面。

根据问题陈述，首先要把本软件所具有的功能呈现给用户，其次需要用户选择排序算法，把输入的数据按照选择的排序算法得到的排序过程过程和排序代码显示给用户。界面具有的属性可以说是排序算法选择按钮，数据框，代码显示框，绘图区，变量值区。类图如下：



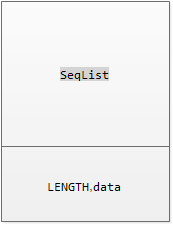
**图4- 5 界面类图**

数据中需要的属性有：坐标、颜色，数值



**图4- 6 数据类图**

数据链中需要的属性有：数据的长度以及数据。



**图**4- 7 **数据链类**

绘图中需要的属性：数据链的对象，界面的对象。

## 4.4 状态模型

### 4.4.1确定状态

状态是对象的生命周期中的一个条件或状况，在此期间对象将满足某些条件、执行某些活动或等待某些时间。当用户启动本软件的界面时软件开始运行，经过显示排序过程和显示排序算法的代码结束本次处理。在这个过程中具有多个状态，首先用户需要启动界面，接着接收用户输入的数据，接受用户的选择，然后显示排序过程和排序算法的代码，下一次开始或结束。软件中有如下状态：开始（用户启动界面）、运行状态（接受用户输入、选择开始处理）结束状态（传递处理结果）。

### 4.4.2构建事件跟踪图

4.4.2.1准备交互式脚本

脚本是指软件在某一次执行期间出现的一系列时间，即序列时间。每当软件进行一次执行时软件中对象与外界用户交换信息即产生一个时间开始，继而产生一系列时间出发整个过程的执行直到结束。

软件一次执行过程的正常脚本：

1. 用户启动软件界面
2. 软件界面要求用户输入待排序的数据
3. 软件界面接受用户输入的数据
4. 软件界面要求用户选择排序算法
5. 软件界面接受用户选择的排序算法，并传递排序算法名进行处理
6. 软件界面接受处理结果
7. 软件界面显示演示过程和用户选择的排序算法的代码
8. 下一次运行或者软件退出

软件一次执行过程中的异常脚本：

1. 用户启动软件界面
2. 软件界面要求用户输入数据
3. 软件界面要求用户选择排序算法
4. 软件界面结束用户选择排序算法
5. 软件界面确认出错，软件内部处理出错
6. 软件界面取消此次操作
7. 软件界面请求用户输入数据
8. 软件界面请求选择排序算法

4.4.2.2 确定事件

根据脚本确定如下时间：启动界面、要求输入、要求选择、接受选择、传递排序算法鸣、接受处理结果、显示演示过程和用户选择的排序算法的代码、等待下一次运行或软件退出等。

4.4.2.3事件跟踪图

从上一节中找出了该软件在一次执行过程中的所发生的事件，和这些时间相关的对象为用户、界面、演示过程、代码等。那么此节的事件跟踪图就是 把这些事件和相应的对象清晰的表达出来，竖线代表对象，箭头代表发生的时间。事件跟踪图如下图所示



**图4- 8软件事件跟踪图**

### 4.4.3构建状态图

下面个给出了排序算法演示的状态图



**图4- 9软件的状态图**

## 4.5服务分析

### 4.5.1服务分析

确定服务需要从两个方面进行:一是类实体的常规行为，二是软件中特殊需要的任务

1. 常规行为

直接从类中导出操作，一般类都定义了属性，而每个属性都是可以被访问的，即每个含有属性的类应该定义能够访问该类属性的服务。界面类含有四个属性，定义访问排序算法名的服务，动画演示的服务，代码显示的服务以及变量值显示的服务。

1. 从状态模型中导出操作

本章节确定了如下时间：要求输入、接受输入、要求选择、接受选择、传递排序算法名、接受处理结果、显示演示过程和排序算法的代码、软件退出。我们可以从中导出如下服务：选择排序算法、传递排序算法名、显示演示过程和排序算法的代码、软件退出等服务。

1. 从交互模型中导出操作

顺序图描述了对象之间动态的交互关系，并且描述了这些对象随时间的推移相互之间交互信息的过程。对象之间的交互信息必须启动相应的操作来完成。我们可以从顺序图得出如下服务：显示界面、输入数据、选择排序算法、传递排序算法明、显示演示过程和排序算法的代码、关闭界面等。

我们从活动图中得到如下操作：输入数据、选择排序算法、对数据进行排序 处理。对数据进行移动处理、排序算法进行显示处理、显示排序演示过程、显示排序算法的代码。

1. 确定服务

经过对得到的服务进行筛选和合并相同服务，得出如下最终服务：获得排序算法明，即获得用户选择的按钮，代码显示服务，动画演示服务。

### 4.5.2软件最终类

经过以上分析我们得出了类以及类的属性和操作，下面对各类进行描述和建立含属性、操作的类。

4.5.2.1类描述

* + 1. 界面类

属性：排序选择按钮、代码显示区、动画演示区

服务：获取排序算法名，显示排序过程，显示排序算法代码

作用：接受用户选择的排序算法名、显示排序演示过程、显示排序算法 代码、显示数值、调用数据链类的数据等。

* + 1. 数据类

属性：坐标X、Y、颜色、数值

服务：颜色、数值、坐标

作用：存储或获取数据的颜色、数值、坐标等

* + 1. 数据链类

属性：数据表，数据的形状大小，链表长度等

服务：数据，坐标

作用：随机产生数据，并计算坐标，并保存数据

* + 1. 绘图类

属性：界面类的对象，数据链类的对象

服务：类的对象

作用：主要用界面类的对象进行绘画显示，调用数据链类的数据。

## 4.6 类的设计

在分析模型中找出的类有如下特点：

1. 该类是根据软件所具备的功能得到的比较理想的类
2. 类中的属性和操作也只是实现软件功能的一部分
3. 类之间的联系比较模糊不利于实际中编码。针对这些特点本节继续分析模型找到的类进行优化以及编码

4.6.1对类进行编码化

对分析模型的前一节中添加的类进行程序化

1. 界面类（SortAlgorithmApplet）

属性：CanvasPanel //绘图 区

Sort\_choice //选择排序按钮

btnDemo //演示按钮

DataPanel //数据区

SourceData//代码区

varPanel //数值区

作用：可以获取数据链中的数据，用户也可以在该类中实现算法选择，以及 本类中也可以控制动画演示、代码同步显示、数值的同步展示

操作：initComponents() //初始化组件

btnDemo\_actionPerformed()//演示开始

Sort\_choice\_Changed（）//选择排序算法

addSourceCode （）//显示代码

insertSort（）//插入排序

bubbleSort（）//冒泡排序

selectSort（）//选择排序

quickSort（）//快速排序

showKeyVlaue（）//显示数值

SelectSource（）//同步显示代码

1. 数据类（Node）

属性：x,y,width,height,value(string),

FillColor,RectColor,ValueColor

作用：主要是用来保存数据的坐标，颜色，数值，同时也是来提取数据的数 据大小，坐标和颜色的。

操作：getX（）

getY（）

getValue（）

setX（）

setY（）

SetValue（）

setFillColor（）

setRectColor（）

setValueColor（）

getFillColor（）

getRectColor（）

getValueColor（）

1. 数据链类（SeqList）

属性：LENGTH，data[]（随机数据产生）

作用：主要是用来保存数据和随机产生数据的类。

操作：getLength（）

getData（）

setData(int[] data)

getData（）

setLength

setData(int index,int e)

1. 绘画类（DemoCanvas）

属性：NodeY，xnode[](主要是用来保存各个数据节点的一组元素)

作用：主要是用来绘图，不断刷新页面从而产生动画效果。

操作：DrawNode（）//画一个数据节点

draw（）//绘画每个数据节点

compare（）//比较两个节点的动画效果

### 4.6.2类之间的联系

首先进入界面类（SortAlgorithmApplet），调用init()函数，进行数据初始化，进行控件初始化initComponents（），添加事件监听事件addListener（）。在进行数据初始化的同时就已经调用了数据链类（SeqList）中的数据元素。用户可在界面类进行选择排序算法名。用户选择排序算法名之后，用flag标记到了用户所选择的算法，如果点击了演示按钮，那么就会进行相应的算法演示，并开启线程进行界面刷新。如下列代码所示

if (this.flag == 0) {

new SortThread(this, 0).start();

} elseif (this.flag == 1) {

new SortThread(this, 1).start();

} elseif (this.flag == 2) {

new SortThread(this, 2).start();

} elseif (this.flag == 3) {

new QuickSortThread(this).start();

}

上面所示，现成也分为两种，一种是为了快速排序单独开的一个线程，一种是另外其他三种排序共用的线程。

首先来讲一下其他三种排序共用的线程SortThread，该线程需要两个参数来判定是那种排序方法，从而调用界面类中的insertSort，bubbleSort，selectSort方法。传入参数flag,如下所示

if (flag==0) {

this.applet.step = this.applet.insertSort(this.applet.step);

}elseif(flag == 1) {

this.applet.step = this.applet.bubbleSort(this.applet.step);

}elseif(flag == 2){

this.applet.step = this.applet.selectSort(this.applet.step);

}

开启线程后，开始调用界面类中的排序方法，下面针对insertSort方法进行讲解。

publicint insertSort(int step) {//插入排序的过程，传入参与step,即执行的步骤

if (step == 0)

this.SourceData.requestFocus();

this.SelectSource(this.insertSortsSrcCode, step); //进行代码的同步选择

switch (step) { //进行到某一步，然后进行某一步的操作

case 1:

countSwich = 0;//第一步，交换计数为0

break;

case 2:

index = indexOfInsertSort - 1;//第二步索引值初始化

break;

case 3: //第三步设定关键值，并同步显示数值表

key = Integer.*parseInt*(this.dc.xnode[indexOfInsertSort].getValue());

showKeyVlaue();

break;

case 4: //进行插入比较，从后往前

if (index >= 0) {

showKeyVlaue();

dc.compare(index, index + 1);

if (Integer.*parseInt*(this.dc.xnode[index].getValue()) < key) {

step = 8;

}

} else {

step = 8;

}

break;

case 5: //交换数值，并刷新画面

this.dc.xnode[index + 1].SetValue(this.dc.xnode[index].getValue());

this.dc.draw();

break;

case 6:

this.dc.xnode[index].SetValue("" + key);

this.dc.draw();

break;

case 7:

countSwich++;

showKeyVlaue();

index--;

step = 3;

break;

case 8:

break;

case 9://表示排序完成，弹出提示框

indexOfInsertSort++;

if (indexOfInsertSort >= SeqList.*LENGTH*) {

String msg = "插入排序算法演示完毕，已排序成功！";

javax.swing.JOptionPane.*showMessageDialog*(null, msg);

return -1;

} else {

step = 1;

}

break;

default:

break;

}

return ++step;

}

线程针对每一步都会刷新，保证排序过程的有序进行，如果返回-1值即表明排序完成，中止排序关闭线程。

### 4.6.3软件运行流程图



**图4- 10软件运行的流程图**

## 4.7模块的详细设计

由软件的总体设计可以得出，软件的四个模块为输入模块、选择模块、处理模块、显示模块。

### 4.7.1输入模块

该模块主要的动作是用户要进行输入数据的长度。如果用户没有进行输入的话，那么点用户点击演示按钮的时候，就会弹出一个消息，提示用户没有进行输入数据长度。这个功能还没有实现，只是自己将要添加的功能，现在只是进行

数据的随机初始化。需要在SeqList类中进行初始化。

代码如下：

public SeqList() {

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

data[i]=(int)(java.lang.Math.*random*()\*100+1);

}

data[length]=-1;

pos=0;

elem=(int)(java.lang.Math.*random*()\*100+1);

}

流程图如下：



**图5- 2输入模块流程图**

### 4.7.2选择模块

该模块主要是负责排序算法的选择，其中有四个选择：插入排序、冒泡排序、选择排序、快速排序。选择的同时就会标记当前选择的是哪种算法以及进行不同的显示，同时界面类也会刷新，绘图区、代码区、数值表区都会跟你选择算法变化而变化。

代码如下：

private void Sort\_choice\_Changed(ItemEvent arg0) {

// TODO Auto-generated method stub

this.dc.repaint();

if (arg0.getItem().equals("插入排序")) {

System.out.println(arg0.getItem());

flag = 0;

this.cardlayout.show(this.varPanel, insertSortName);

this.addSourceCode(this.insertSortsSrcCode);

} else if (arg0.getItem().equals("冒泡排序")) {

System.out.println(arg0.getItem());

this.cardlayout.show(this.varPanel, bubbleSortName);

flag = 1;

this.addSourceCode(this.bubbleSorthSrcCode);

} else if (arg0.getItem().equals("选择排序")) {

System.out.println(arg0.getItem());

this.cardlayout.show(this.varPanel, selectSortName);

flag = 2;

this.addSourceCode(selectSortSrcCode);

} else {

System.out.println(arg0.getItem());

this.cardlayout.show(this.varPanel, quickSortName);

flag = 3;

this.addSourceCode(quickSortSrcCode);

}

this.dc.draw();

}

流程图如下：



**图5- 3选择模块的流程图**

### 4.7.3处理模块

该模块主要是控制用哪种排序演示。对以上两个模块存储的数据进行处理。 首先要进行判断是哪种算法，通过上面选择模块保存的数据就可以进行判断，点击演示按钮，调用处理动作方法，代码如下：

public void btnDemo\_actionPerformed(ActionEvent e) {

step = 0;

indexOfInsertSort = 1;

indexOfBubbleSort = 0;

indexOfSelectSort = 0;

a = new int[] { 0, SeqList.LENGTH-1 };

if (list.size() == 0) {

list.add(a);

}

this.dc.init();

this.dc.draw();

btnDemo.setEnabled(false);

// this.CheckVar();

if (this.flag == 0) {

new SortThread(this, 0).start();

} else if (this.flag == 1) {

new SortThread(this, 1).start();

} else if (this.flag == 2) {

new SortThread(this, 2).start();

} else if (this.flag == 3) {

new QuickSortThread(this).start();

}

}

其次进行数据的加载处理，由上述代码可知，加载的线程也要传输参数，进而判断是哪种排序算法。在线程中，同时也会判断终止条件，以方便进行线程处理，中止排序。

public void run() {

// TODO Auto-generated method stub

while (true) {

try {

sleep(sleeptime);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if (flag==0) { //判断flag标记,选择排序算法的执行方式

this.applet.step = this.applet.insertSort(this.applet.step);

}else if(flag == 1) {

this.applet.step = this.applet.bubbleSort(this.applet.step);

}else if(flag == 2){

this.applet.step = this.applet.selectSort(this.applet.step);

}

if (this.applet.step == -1) { //排序算法的中止条件就是返回值为-1

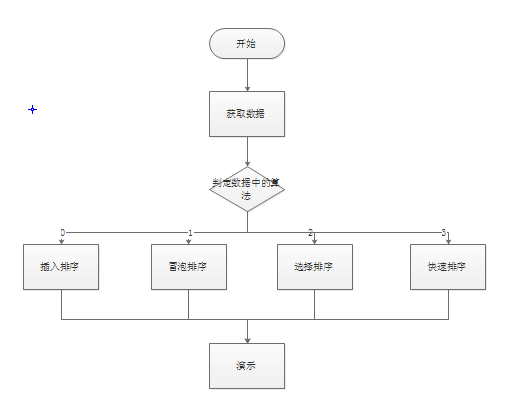
this.applet.btnDemo.setEnabled(true);

break;

}

}

}



**图5- 4处理模块流程图**

### 4.7.4显示模块

显示模块主要是对动画显示以及代码同步和数据同步，也是为四个算法进行选择。

动画显示主要是在CanvasDemo中进行绘图

代码如下:

public void paint(Graphics g) {

if (slda.flag == 3) {

for (int i = 0; i < sl.getLength()+1; i++) {

this.DrawNode(i, g);

}

}else{

for (int i = 0; i < sl.getLength(); i++) {

this.DrawNode(i, g);

}

}

}

这个只是进行总的绘画，还进行了节点绘画，节点绘画也绘画文字，方框。

代码如下:

public void DrawNode(int index, Graphics g) {

g.setColor(xnode[index].getFillColor());

g.fillRect(xnode[index].getX(), xnode[index].getY(), Node.width,

Node.height);

g.setColor(xnode[index].getRectColor());

g.drawRect(xnode[index].getX(), xnode[index].getY(), Node.width,

Node.height);

g.setColor(xnode[index].getValueColor());

g.drawString(String.valueOf(xnode[index].getValue()),

xnode[index].getX() + Node.width / 2, xnode[index].getY()

+ Node.height / 2);

}

代码同步，主要是传入一个参数step，然后进行选择同一步骤的代码。

代码如下:

public void SelectSource(String src[], int row) {

int start = 0;

if (row > src.length - 1)

row = src.length - 1;

for (int i = 0; i < row; i++)

start += (src[i].length() + 1);

this.SourceData.select(start, start + src[row].length());

}

数值同步，主要还是每次数值的变化，都进行对数值表进行刷新，调用刷新数值的方法。

代码如下：

public void showKeyVlaue() {

if (flag == 0) {

this.v\_n.setText("" + SeqList.LENGTH);

this.v\_i.setText("" + index);

this.v\_key.setText("" + key);

this.v\_j.setText("" + indexOfInsertSort);

this.v\_compare.setText("" + this.dc.xnode[index].getValue());

this.v\_countSwitch.setText("" + this.countSwich);

} else if (flag == 1) {

this.v\_n1.setText("" + SeqList.LENGTH);

this.v\_i1.setText("" + index);

this.v\_datai.setText(this.dc.xnode[index].getValue());

this.v\_j1.setText("" + indexOfBubbleSort);

this.v\_dataj.setText(this.dc.xnode[indexOfBubbleSort].getValue());

this.v\_countSwitch1.setText("" + countSwich);

} else if (flag == 2) {

this.v\_n2.setText("" + SeqList.LENGTH);

this.v\_i2.setText("" + index);

this.v\_lowestIndex.setText("" + indexOfKey);

this.v\_datalowestIndex

.setText(this.dc.xnode[indexOfKey].getValue());

this.v\_j2.setText("" + indexOfSelectSort);

this.v\_dataj2.setText(this.dc.xnode[indexOfSelectSort].getValue());

this.v\_countSwitch2.setText("" + countSwich);

}else {

this.v\_n3.setText(""+SeqList.LENGTH);

this.v\_x.setText(""+x);

this.v\_l.setText(""+l);

this.v\_r.setText(""+r);

this.v\_i3.setText("" + i);

this.v\_datai3.setText(this.dc.xnode[i].getValue());

this.v\_j3.setText("" + j);

this.v\_dataj3.setText(this.dc.xnode[j].getValue());

this.v\_countswitch3.setText(""+countSwich);

}

}

# 第5章 软件实现和测试

在分析和设计阶段已经明确软件应该做哪些内容，本章主要是把应该做的事情实现出来，并进行测试。

## 5.1软件实现

### 5.1.1软件设计

5.1.1.1软件体系结构设计

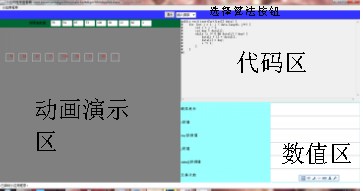
软件通过用户选择排序算法开始执行，值能由用户激发软件开始执行，一次只能处理一种排序算法，一次处理完之后，然后软件将处理的结果进行反馈给用户。由用户和软件交互来支配软件的运行。下图给出了软件的交互图：



**图5- 7软件的交互图**

5.1.1.2界面设计

本软件功能简单，没有高复杂性。只要求能够使选择排序算法将最终的排序演示和排序算法的代码显示出来，需要用户操作的只有选择排序算法，之后就可看到结果。本软件需要有选择排序算法的按钮、代码区、数值区、以及动画演示区。



**图5- 8软件的界面截图**

## 5.2软件测试

### 5.2.1测试环境

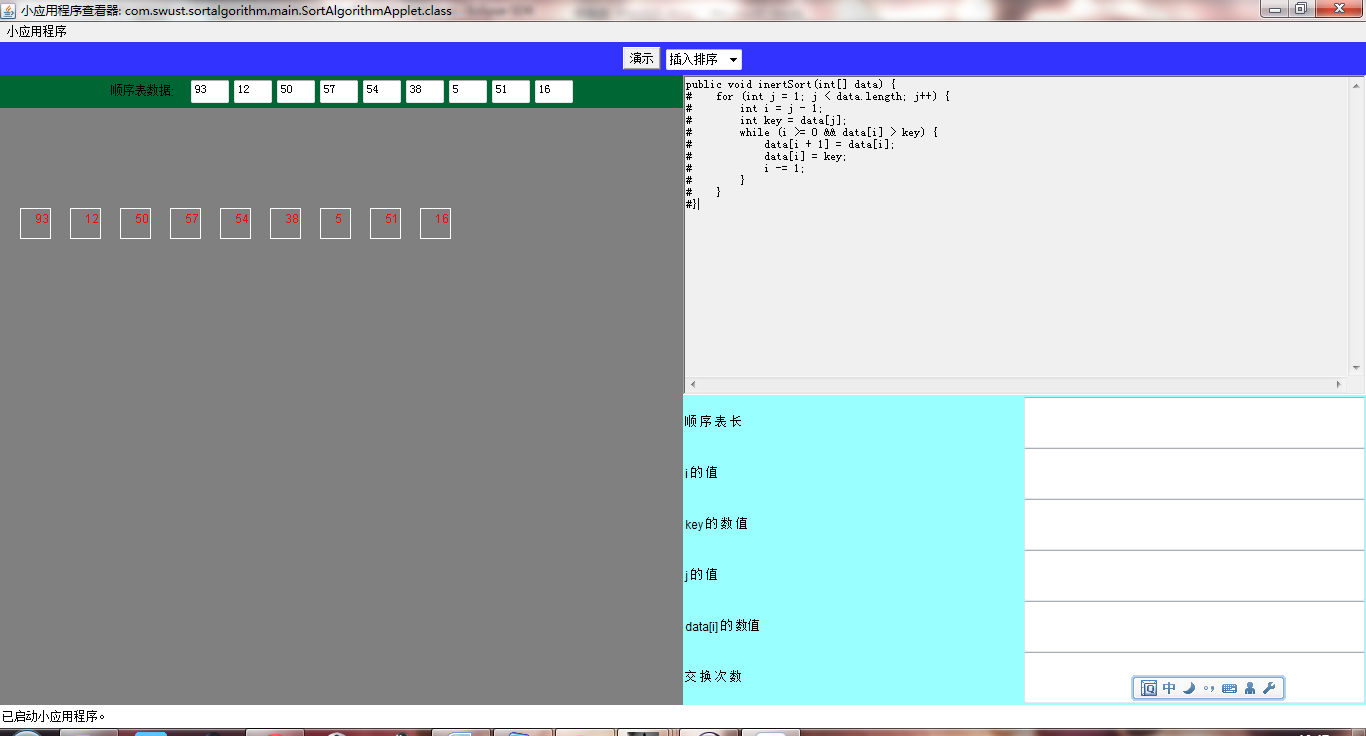
软件测试环境：WINXP/WIN7/WIN8操作软件，编程工具eclipse，装有JDK6.0 以上接口

硬件测试环境：内存要求在512M及其以上奔腾系列机

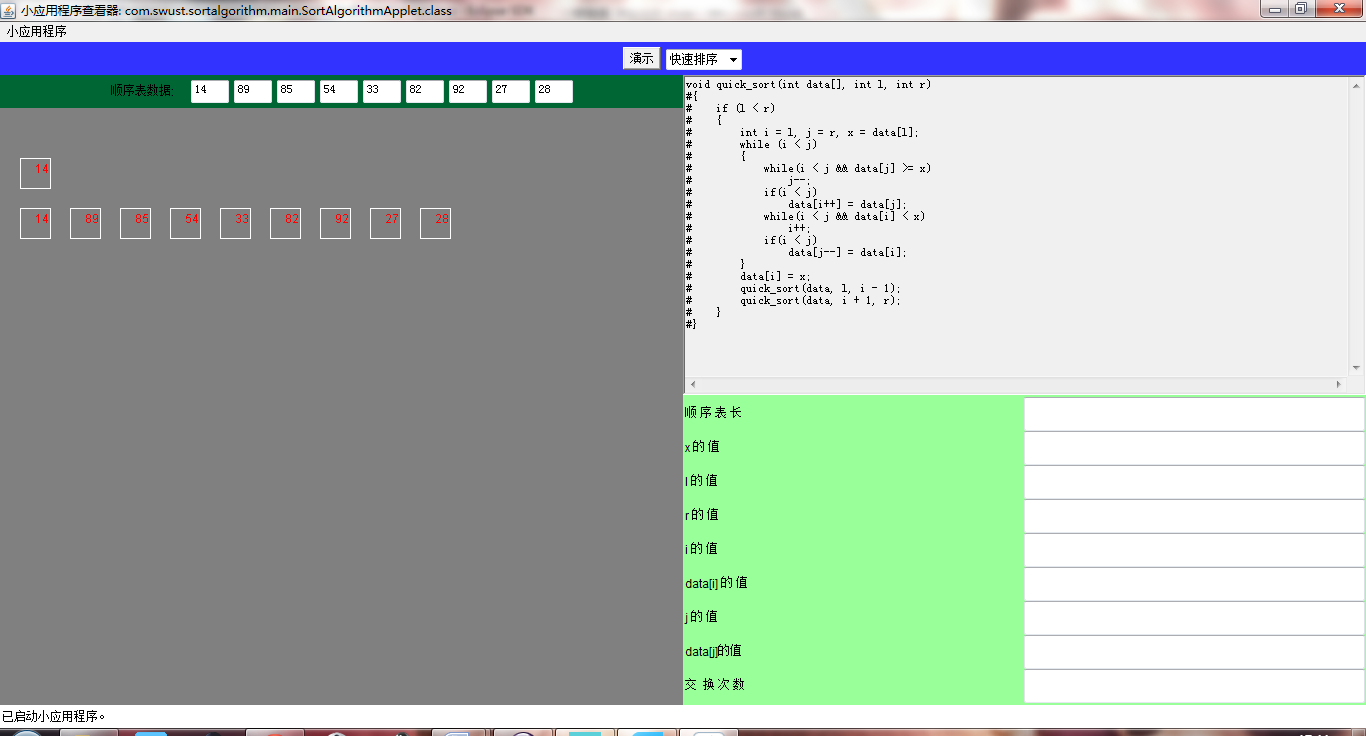
### 5.2.2用户界面测试

用户界面是用户和软件的接口，一个好的用户界面不仅要求能够方便用户使用而且美观。所以用户界面测试的主要任务是测试用户界面是否友好，界面上的各个控件是否好用

启动应用程序，选择一个排序算法对界面进行测试，测试界面未发现不友好现象。



**图5- 9测试界面（一）**



**图5- 10测试界面（二）**

### 5.2.3简单功能测试

本节主要是对软件的功能进行简单测试，按照功能名称、操作、结果来分类。

**表5- 1简单的功能测试表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 功能名称 | 操作 | 结果 |
| 1 | 输入数据长度 | 无 | 还没有实现这个功能 |
| 2 | 选择排序算法 | 点击选择按钮即可 | 出现四种算法可供选择：插入排序、冒泡排序、选择排序、快读排序 |
| 3 | 启动排序算法演示 | 点击演示按钮 | 成功启动排序演示 |
| 4 | 代码同步 | 无 | 在启动演示后，代码同步显示 |
| 5 | 数值同步 | 无 | 在启动演示后，代码同步显示 |
| 6 | 演示冒泡排序算法 | 点击选择算法按钮，选择冒泡排序算法 | 显示冒泡排序算法，进行了动画演示，以及数值更新同步 |
| 7 | 演示插入排序算法 | 点击选择算法按钮，选择插入排序算法 | 显示插入排序算法，进行了动画演示，以及数值更新同步 |
| 8 | 演示选择排序算法 | 点击选择算法按钮，选择选择排序算法 | 显示选择排序算法，进行了动画演示，以及数值更新同步 |
| 10 | 排序完毕提示 | 无 | 排序完成后，提示排序已完成 |

# 结论和展望

## 完成的任务

* 1. 实现内排序算法中的插入排序，冒泡排序，选择排序，快速排序等四个算法的演示。
  2. 用绘图的方式实现动态演示
  3. 给出代码区,程序执行步骤和演示步骤能由代码区反映
  4. 显示关键变量的值变化情况

## 收获总结

对本软件的开发，可以看出各个不同的排序算法排序的过程，以及之间在排列整数时的差异。这四种算法中，快速排序比较和移动的次数是最少的，也是最快的一种排序方法。直接选择排序虽然交换次数很少，但是比较次数较多。冒泡排序应该算是里面最慢的一种排序算法了，不仅比较多而且交换次数也多，但是最为简单且好理解。

通过毕业设计的锻炼，学到了很多东西。首先是巩固和加深了对数据结构的理解，提高综合运用所学知识的能力。其次是培养了我选用参考书，自己动作查阅手册及文献资料的能力，培养自己独立思考，深入研究、分析问题、解决问题的能力。再次通过自己编写的程序对各种排序性能的比较让我更深入理解了它们的应用。然后按照毕业设计的要求来完成毕设报告书，能正确阐述设计和实验结果，正确绘制软件结构图、顺序图、活动图等。最后也是通过这次毕设让自己的对软件开发整个流程熟悉了一遍，从无到有，从设计到开发。同时也让我感觉很多地方不足，java中的东西懂的还是皮毛，尤其是对线程方面的匮乏，也发现自己软件工程学的很差，很多地方都无从下手。而且由于自己的马虎和大意，导致了浪费了很多时间。通过此次设计，也让我更加严谨对待自己的学习，逐步建立正确的全局观念。毕业设计是把我们所学的理论知识进行软件的总结并应用于实践的良好机会，有利于加强我们用知识理论来分析实际问题的能力，进而加强了我们对知识认识的实践度，巩固我们的理论只是，神话了对知识的认识，并为我们走向社会打下了一下良好的基础。

## 展望未来

本软件大致实现了功能需求，但是本人能力确实有限，还有很多不足有待自己解决，下面主要对未来这个软件进行的扩展进行简单陈述：

1. 由于底子比较薄，所以软件设计不够好看，还有很多值得改进的地方，比如软件的性能方面，以及代码构造方面和界面优化等等
2. 软件设计可以进行扩展，增加一个日志输出，可以针对每个步骤在做什么进行一个简要说明。

# 致谢

毕业论文到此结束，这也意味着我在西南科技大学的四年学习生活也即将结束。回首过去，发现这四年来收获了很多情谊，许多知识，许多人生哲理。在西南科技大学这四年中，思想、生活、学习等等都得到了很大的改善及提高。所以在这四年中要感谢的人很多很多，在这里对哪些一直包容、帮助我的人来说声谢谢。四年大学，感谢有你，感谢有你们。

非常感谢我的指导老师孙敏老师在我毕业设计阶段给予的帮助。从刚开始毕设不会到毕设的完成。也感谢他在我的毕业论文从最初的定稿，到资料搜集，到写作、修改到最后论文定稿，他都给予了我耐心的指导和无私的帮助。从而让我顺利的完成自己的毕业设计。

感谢所有任课老师在这四年的教导，你们不仅教会了我专业知识，还指导着我怎么去为人处事，如何学习。

感谢我们班上的全体同学，无论在学习上、生活上都给予了我莫大的鼓励和支持，同我一起成长奋斗。

在这里也感谢自己，坚持了四年自己要过的生活，从来没有放弃过自己想要的生活，也一直锻炼自己的耐心和保持一种乐观心态，对自己以后在社会上行走有很大的帮助。

# 参考文献

[1]朱战立.数据结构 使用C语言[M].（第三版）.西安：西安交通大学出版社，2004.234-257

[2]Bruce Eckel .Java编程思想[M].（第四版）.北京：机械工业出版社，2007.6.669-672

[3]Ian Sommerville.软件工程[M].（第八版）.北京：机械工业出版社，2006.9.245-260

[4]王克宏、董丽、朱家维.Java技术及其应用[M].（第一版）.北京：高等教育出版社，1999.125-138

[5]张海藩.软件工程导论[M].（第三版）.北京：清华大学出版社，1998.38-41

[6]Mark Allen Weiss.数据结构与算法分析[M].（第二版） .北京：机械工业出版社，2008.3.183-195

[7]张曜、张青、郭立山.Java程序设计教程[M].（第一版）.北京：清华大学出版社，2002.11.38-41

[8]宋雨、赵文清. 软件工程[M].（第一版）.北京：中国电力出版社，2007.182-198

[9] 印旻.[Java与面向对象程序设计教程](http://d.wanfangdata.com.cn/ExternalResource-bjlhdxxb200202013^2.aspx)[M].北京:高等教学出版社,2001.

[10]张卫民,黄瑞芳,卢宇彤. [Java语言及应用](http://d.wanfangdata.com.cn/ExternalResource-bjlhdxxb200202013^3.aspx)[M].北京:清华大学出版社,1997.

[11]MacoJack. [Particle Java Data Structure](http://d.wanfangdata.com.cn/ExternalResource-nchkgyxyxb200602018^5.aspx)[M].Wrox Publish,1996.

[12]Wolf gang Christian. [Java programming and Internet technologies for underg-raduate education](http://d.wanfangdata.com.cn/ExternalResource-nchkgyxyxb200602018^6.aspx)[J].Computer Physics Communications,2000,(01).