**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**🙠**🕮**🙢**



**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC NGÔN NGỮ LẬP TRÌNH JAVA**

**ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH MÔ PHỎNG CÁC THUẬT TOÁN SẮP XẾP**



Lớp: SE330.I21.PMCL

GVHD: Huỳnh Tuấn Anh

Nhóm sinh viên thực hiện:

1. Phan Vĩnh Huy – 15520315

**TP. Hồ Chí Minh, 7/2019**

# **LỜI CẢM ƠN**

Để hoàn thành tốt đề tài này chúng em xin trân trọng cảm ơn Thầy **Huỳnh Tuấn Anh -** giảng viên hướng dẫn môn học của lớp Ngôn Ngữ Lập Trình Java (*SE330.I21.PMCL*) đã tận tình giúp đỡ, chỉ bảo hướng dẫn chúng em trong suốt quá trình thực hiện đồ án. Chúng em cũng cảm ơn thầy đã giảng dạy chúng em, những kiến thức trên giảng đường là hành trang giúp chúng em vững bước trong tương lai.

Vì thời gian và năng lực còn có hạn chế nên không thể tránh khỏi những sai sót trong khi thực hiện đề tài. Rất mong được sự góp ý bổ sung của thầy để đề tài của chúng em ngày càng hoàn thiện hơn.

Xin chân thành cảm ơn Thầy!

# **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Mục Lục**

[**LỜI CẢM ƠN** 2](#_Toc13169788)

[**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN** 3](#_Toc13169789)

[**I. MỞ ĐẦU** 6](#_Toc13169790)

[**I.1 Lý do chọn đề tài** 6](#_Toc13169791)

[**I.2 Mục tiêu và nhiệm vụ** 6](#_Toc13169792)

[**II. TỔ CHỨC VÀ QUẢN LÝ DỰ ÁN** 6](#_Toc13169793)

[**II.1 Cấu trúc tổ chức** 6](#_Toc13169794)

[**II.2 Nơi lưu trữ và quản lý mã nguồn chương trình** 7](#_Toc13169795)

[**III. CƠ SỞ LÝ THUYẾT VỀ CÁC THUẬT TOÁN SẮP XẾP** 7](#_Toc13169796)

[**III.1 Tổng quan** 7](#_Toc13169797)

[**III.2 Các thuật toán sắp xếp** 7](#_Toc13169798)

[***III.2.1 Thuật Toán Selection Sort*** 7](#_Toc13169799)

[***III.2.2 Thuật Toán Insertion Sort*** 7](#_Toc13169800)

[***III.2.3 Thuật Toán Bubble Sort*** 8](#_Toc13169801)

[***III.2.4 Thuật Toán Interchange Sort*** 8](#_Toc13169802)

[***III.2.5 Thuật Toán Shell Sort*** 9](#_Toc13169803)

[***III.2.6 Thuật Toán Heap Sort*** 10](#_Toc13169804)

[***III.2.7 Thuật Toán Quick Sort*** 10](#_Toc13169805)

[***III.2.8 Thuật toán Merge Sort*** 11](#_Toc13169806)

[**IV. THIẾT KẾ GIAO DIỆN** 12](#_Toc13169807)

[**IV.1 Giao diện chính của chương trình** 12](#_Toc13169808)

[**IV.2 Khung chạy mô phỏng** 12](#_Toc13169809)

[**IV.3 Giao diện chức năng nhập và thao tác dữ liệu trên mảng** 13](#_Toc13169810)

[**IV.4 Khung hiển thị Code C/C++ của các thuật toán sắp xếp** 13](#_Toc13169811)

[**IV.5 Khung hiển thị nội dung ý tưởng thuật toán** 14](#_Toc13169812)

[**IV.6 Khung các chức năng điều khiển quá trình mô phỏng** 14](#_Toc13169813)

[**V. CÀI ĐẶT** 15](#_Toc13169814)

[**V.1 Tổng quan về ngôn ngữ lập trình JAVA** 15](#_Toc13169815)

[**V.2 Cách cài đặt chương trình** 16](#_Toc13169816)

[**VI. THIẾT KẾ CHỨC NĂNG** 18](#_Toc13169817)

[**VII. KIỂM THỬ** 19](#_Toc13169818)

[**VII.1 Qui trình kiểm thử:** 19](#_Toc13169819)

[**VII.2 Kết quả kiểm thử** 20](#_Toc13169820)

[**VIII. NHẬN XÉT VÀ KẾT LUẬN** 22](#_Toc13169821)

[**VIII.1. Đánh giá tổng quan** 22](#_Toc13169822)

[**VIII.2 Ưu điểm và nhược điểm** 22](#_Toc13169823)

[***VIII.2.1. Ưu điểm*** 22](#_Toc13169824)

[***VIII.2.2. Nhược điểm*** 22](#_Toc13169825)

[***VIII.2.3. Hướng phát triển*** 22](#_Toc13169826)

[**TÀI LIỆU THAM KHẢO** 23](#_Toc13169827)

# **I. MỞ ĐẦU**

## **I.1 Lý do chọn đề tài**

Các thuật toán sắp xếp rất quan trọng trong ngành công nghệ thông tin nói chung và môn cấu trúc dữ liệu và giải thuật nói riêng. Đó là kiến thức nền tảng không thể thiếu đối với sinh viên ngành công nghệ thông tin. Để phục vục như một tài liệu hướng dẫn trong việc dạy các thuật toán sắp xếp bằng máy tính, đặc biệt giúp học sinh và sinh viên hiểu và hình dung quá trình thay đổi ở cấu trúc dữ liệu của các thuật toán sắp xếp trong suốt quá trình nó thực thi trở nên nhanh hơn, dễ dàng hơn vì vậy nhóm chọn đề tài **xây dựng công cụ mô phỏng các thuật toán sắp xếp**.

## **I.2 Mục tiêu và nhiệm vụ**

- Nghiên cứu tổng quan về các thuật toán sắp xếp hiện nay

- Hướng đến các kỹ thuật lập trình với ngôn ngữ lập trình Java

- Áp dụng kết quả đã tìm hiểu làm một Demo chương trình mô phỏng các thuật toán sắp xếp

# **II. TỔ CHỨC VÀ QUẢN LÝ DỰ ÁN**

## **II.1 Cấu trúc tổ chức**

|  |  |  |
| --- | --- | --- |
| MSSV | Họ tên | Vai trò |
| 15520315 | Phan Vĩnh Huy | Developer |

## **II.2 Nơi lưu trữ và quản lý mã nguồn chương trình**

Mã nguồn chương trình Demo được lưu trữ và quản lý bằng Github ở đường link sau:

<https://github.com/phanvinhhuy/MoPhong_Sorts>

# **III. CƠ SỞ LÝ THUYẾT VỀ CÁC THUẬT TOÁN SẮP XẾP**

## **III.1 Tổng quan**

Sắp xếp là một quá trình biến đổi một danh sách các đối tượng thành một danh sách thỏa mãn một thứ tự xác định nào đó. Sắp xếp đóng vai trò quan trọng trong tìm kiếm dữ liệu.

## **III.2 Các thuật toán sắp xếp**

### ***III.2.1 Thuật Toán Selection Sort***

**- Ý tưởng**: Xuất phát từ đầu dãy, tìm tất cả các cặp nghịch thế chứa phần tử này, triệt tiêu chúng bằng cách đổi phần tử này với phần tử tương ứng trong cặp nghịch thế. Lặp lại xử lý trên với các phần tử tiếp theo.

**- Các bước thực hiện thuật toán:**

● Bước 1:   i = 0

● Bước 2:  Tìm phần tử **a[min]** nhỏ nhất trong dãy hiện hành từ a[i] đến a[N]

● Bước 3:  Đổi chỗ a[min] và a[i]

● Bước 4:  Nếu i < N-1 thì i=i+1 🡪 Lặp lại bước 2 - Ngược lại dừng

### ***III.2.2 Thuật Toán Insertion Sort***

**-Ý tưởng**:Giả sử có một dãy a(0),a(1),...,a(n-1) trong đó i phần tử đầu tiên a(0),a(1),...,a(i-1) đã có thứ tự. Tìm cách chèn phần tử a(i) vào vị trí thích hợp của đoạn đã được sắp để có dãy mới a(0),a(1),...,a(i) trở nên có thứ tự. Vị trí này chính là vị trí giữa hai phần tử a(k-1) và a(k) thỏa a(k-1)<a(i)<a(k) (1<=k<=i)

**- Các bước thực hiện của thuật toán:**

● Bước 1: i = 1; //giả sử có đoạn a[1] đã được sắp

● Bước 2: x = a[i]; Tìm vị trí pos thích hợp trong đoạn a[1] đến a[i-1] để chèn a[i] vào

● Bước 3: Dời chỗ các phần tử từ a[pos] đến a[i-1] sang phải 1 vị trí để dành chổ cho a[i]

● Bước 4: a[pos] = x; //có đoạn a[1]..a[i] đã được sắp

● Bước 5: i = i+1 🡪 Nếu i < n: Lặp lại bước 2 – Ngược lại: Dừng

### ***III.2.3 Thuật Toán Bubble Sort***

**- Ý tưởng**: Xuất phát từ cuối dãy, đổi chỗ các cặp phần tử kế cận để đưa phần tử nhỏ hơn hoặc lớn hơn trong cặp phần tử đó về vị trí đúng đầu dãy hiện hành, sau đó sẽ không xét đến nó ở bước tiếp theo, do vậy ở lần xử lý thứ i sẽ có vị trí đầu dãy là i. Lặp lại xử lý trên cho đến khi không còn cặp phần tử nào để xét

**- Các bước thực hiện của thuật toán:**

● Bước 1: i = 0; // lần xử lý đầu tiên

● Bước 2: j = N-1 //Duyệt từ cuối dãy ngược về vị trí i

Trong khi (j > i) thực hiện:

Nếu a[j]<a[j-1]

Doicho(a[j], a[j-1])*;*

j = j-1;

● Bước 3: i = i+1; // lần xử lý kế tiếp

Nếu i >=N-1: Hết dãy. Dừng

Ngược lại: Lặp lại Bước 2.

### ***III.2.4 Thuật Toán Interchange Sort***

**- Ý tưởng**: Xuất phát từ đầu dãy, tìm tất các các nghịch thế chứa phần tử này, triệt tiêu chúng bằng cách đổi chỗ 2 phần tử trong cặp nghịch thế. Lặp lại xử lý trên với phần tử kế trong dãy.

**- Các bước thực hiện thuật toán:**

● Bước 1: i = 0; *// bắt đầu từ đầu dãy*

● Bước 2: j = i+1; *//tìm các nghịch thế với a[i]*

● Bước 3:

Trong khi j < N thực hiện

Nếu a[j]<a[i] *//xét cặp a[i], a[j]*

Swap(a[i], a[j]);

j = j+1;

● Bước 4: i = i+1;

Nếu i < N-1: Lặp lại Bước 2.

Ngược lại: Dừng.

### ***III.2.5 Thuật Toán Shell Sort***

**- Ý tưởng**: Cải tiến phương pháp InsertionSort. Phân hoạch dãy thành các dãy con. Sắp xếp các dãy con theo phương pháp InsertionSort. Dùng phương pháp InsertionSort sắp xếp lại cả dãy. Tìm k bước với các khoảng cách chọn theo công thức h(i) = (h(i-1) - 1)/2 và h(k) = 1, k = lg(n)/lg(2) – 1.

**- Các bước thực hiện thuật toán:**

● Bước 1: Chọn **k** khoảng cách h[1], h[2], ..., h[k];

i = 1;

● Bước 2: Phân chia dãy ban đầu thành các dãy con cách nhau h[i] khoảng cách.

Sắp xếp từng dãy con bằng phương pháp chèn trực tiếp;

● Bước 3: i = i+1;

Nếu i > k: Dừng

Ngược lại: Lặp lại Bước 2.

### ***III.2.6 Thuật Toán Heap Sort***

- Ý tưởng: Khi tìm phần tử nhỏ nhất (lớn nhất) ở bước i, phương pháp SelectionSort không tận dụng được các thông tin đã có được do các phép so sánh ở bước i. Giải thuật HeapSort khắc phục nhược điểm này bằng cách chọn ra được một cấu trúc dữ liệu cho phép tích lũy các thông tin về sự so sánh giá trị các phần tử trong quá trình sắp xếp

- Các bước thực hiện của thuật toán:

● Giai đoạn 1: Hiệu chỉnh dãy số ban đầu thành heap

● Giai đoạn 2: Sắp xếp dãy số dựa trên heap:

Bước1: Đưa phần tử lớn nhất về vị trí đúng ở cuối dãy:

r = n;

Hoánvị (a1, ar);

Bước 2: Loại bỏ phần tử lớn nhất ra khỏi heap: r = r-1;

Hiệu chỉnh phần còn lại của dãy từ a1, a2 ... ar thành một heap.

Bước 3:

Nếu r>1 (heap còn phần tử): Lặp lại Bước 2

Ngược lại: Dừng

### ***III.2.7 Thuật Toán Quick Sort***

**- Ý tưởng:** Sắp xếp dãy a(1),a(2),...,a(n) dựa trên việc phân hoạch dãy ban đầu thành 3 phần :

+ Phần 1: Gồm các phần tử có giá trị bé hơn x.

+ Phần 2: Gồm các phần tử có giá trị bằng x.

+ Phần 3: Gồm các phần tử có giá trị lớn hơn x

(Với x là giá trị của một phần tử tùy ý trong dãy ban đầu)

**- Các bước thực hiện thuật toán:**

● Bước 1: Nếu left ≥ right //dãy có ít hơn 2 phần tử

Kết thúc; //dãy đã được sắp xếp

● Bước 2: Phân hoạch dãy aleft … aright thành các đoạn: aleft.. aj,aj+1.. ai-1,ai.. aright

*Đoạn 1 ≤ x*

*Đoạn 2: aj+1.. ai-1 = x*

*Đoạn 3: ai.. aright* ≥ x

● Bước 3: **Sắp xếp đoạn 1**: aleft.. aj

● Bước 4: **Sắp xếp đoạn 3**: ai.. aright

### ***III.2.8 Thuật toán Merge Sort***

**- Ý tưởng:** Sắp xếp dãy a(1),a(2),...,a(n) dựa trên nhận xét sau:

+ Mỗi dãy a(1),a(2),...,a(n) bất kỳ là một tập hợp các dãy con liên tiếp mà mỗi dãy con đều đã có thứ tự.

+ Dãy đã có thứ tự coi như có 1 dãy con.

Hướng tiếp cận: tìm cách làm giảm số dãy con không giảm của dãy ban đầu

**- Các bước thực hiện thuật toán:**

● Bước 1: k = 1; *// dãy con có 1 phần tử là dãy không giảm*

● Bước 2: Lặp trong khi (k < N) *// dãy còn hơn 1 dãy con*

+ Bước 2.1: Phân phối đều luân phiên dãy a1, a2, …, an thành 2 dãy b, c theo từng nhóm k phần tử liên tiếp nhau.

*//b = a1, …, ak, a2k+1, …, a3k, …*

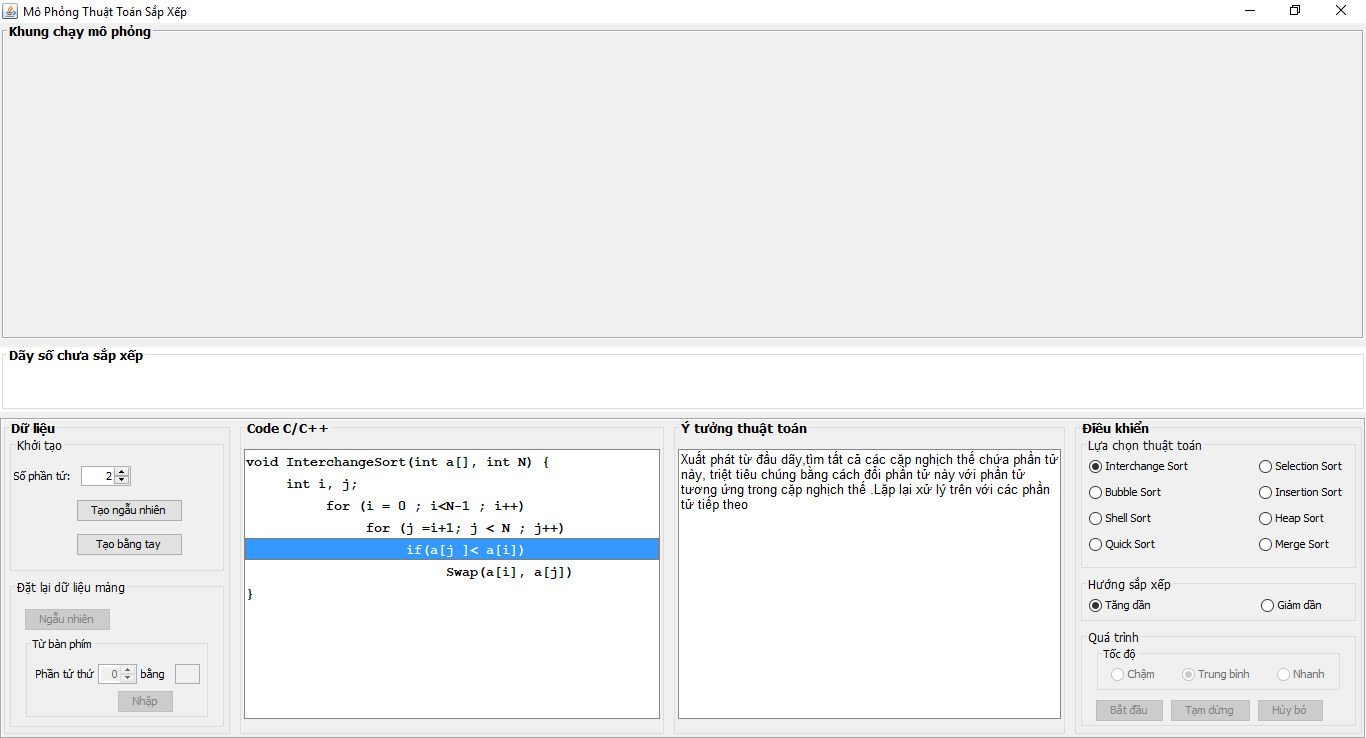
*//c = ak+1, …, a2k, a3k+1, …, a4k, …*

+ Bước 2.2: Trộn từng cặp dãy con gồm k phần tử của 2 dãy b, c vào a.

+ Bước 2.3: k = k\*2;

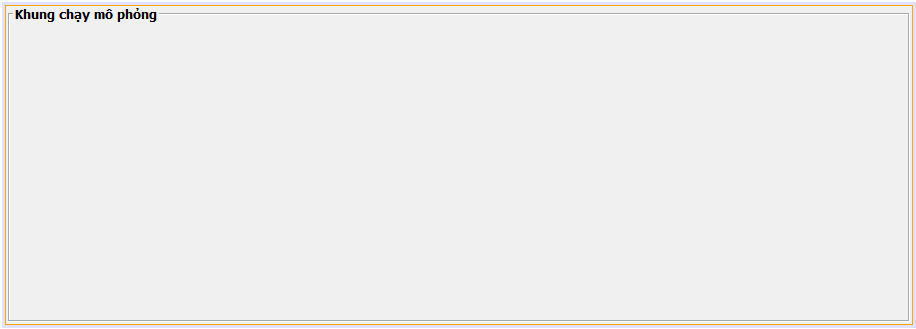
# **IV. THIẾT KẾ GIAO DIỆN**

## **IV.1 Giao diện chính của chương trình**



*Hình 4.1: Giao diện chính của chương trình*

## **IV.2 Khung chạy mô phỏng**

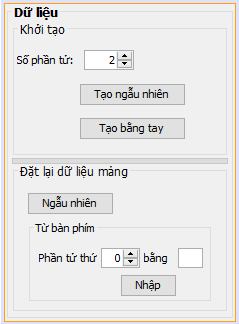


*Hình 4.2: Khung chạy mô phỏng các bước của thuật toán*



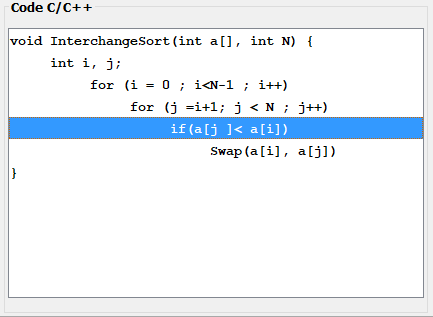
*Hình 4.3: Khung hiển thị dãy số khi chưa được sắp xếp*

## **IV.3 Giao diện chức năng nhập và thao tác dữ liệu trên mảng**



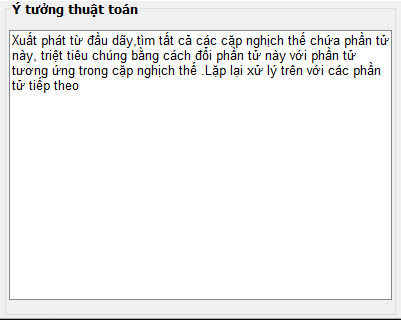
*Hình 4.4: Khung các chức năng nhập và thao tác dữ liệu trên mảng*

## **IV.4 Khung hiển thị Code C/C++ của các thuật toán sắp xếp**



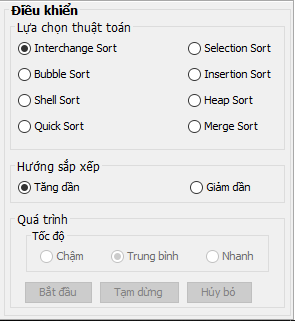
*Hình 4.5: Khung hiển thị Code C/C++ của các thuật toán sắp xếp*

## **IV.5 Khung hiển thị nội dung ý tưởng thuật toán**



*Hình 4.6: Khung hiển thị nội dung ý tưởng thuật toán*

## **IV.6 Khung các chức năng điều khiển quá trình mô phỏng**



*Hình 4.7: Khung các chức năng điều khiển quá trình mô phỏng*

# **V. CÀI ĐẶT**

## **V.1 Tổng quan về ngôn ngữ lập trình JAVA**

- Khái niệm: Java là một ngôn ngữ lập lập trình, được phát triển bởi Sun Microsystem vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.



- Ứng dụng:

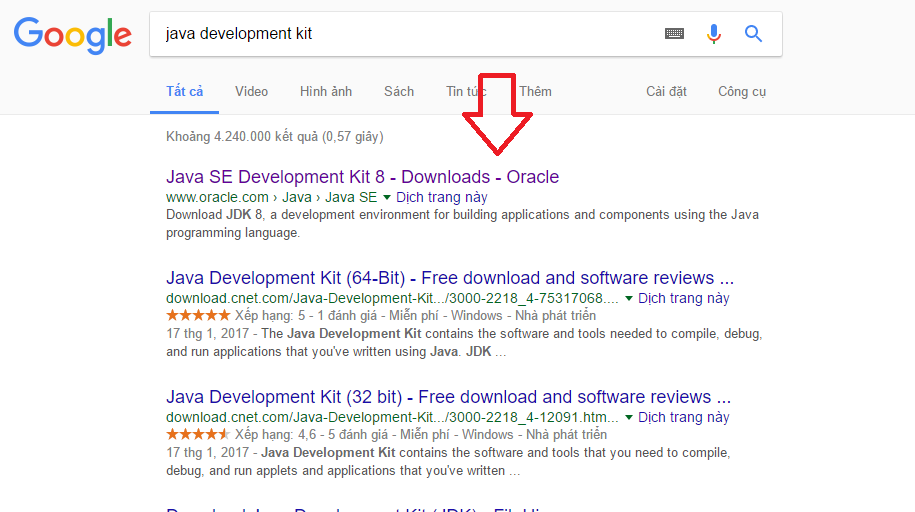
* Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.
* Tạo các trang web có nội dung động (web applet), nâng cao chức năng của server.
* Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí, ...

## **V.2 Cách cài đặt chương trình**

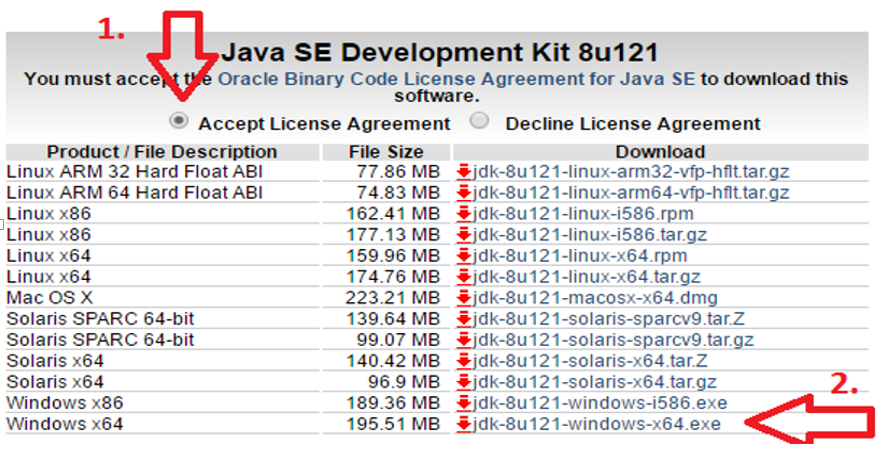
- Chương Trình Mô Phỏng Các Thuật Toán Sắp Xếp được viết bằng ngôn ngữ lập trình Java dựa trên nền tảng Java JDK 1.8

- Cách cài đặt JDK:

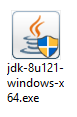
• Bước 1: JDK là viết tắt của Java Development Kit. Người dùng có thể lên google tìm kiếm từ khóa: Java Development Kit và click vào link đầu tiên



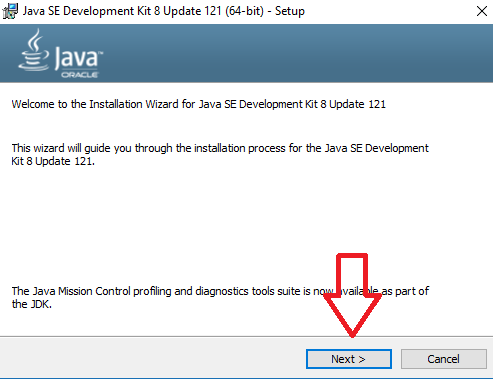
• Bước 2: Các bạn chọn Accept License Agreement và sau đó chọn phiên bản phù hợp với hệ điều hành của mình



• Bước 3: Sau khi tải về máy, các bạn mở file vừa tải về lên và tiến hành cài đặt



• Bước 4: Sau khi mở file, các bạn cài đặt



Các bạn chọn Next 🡪 Next và đợi cài đặt xong

# **VI. THIẾT KẾ CHỨC NĂNG**

Chương Trình Mô Phỏng Các Thuật Toán Sắp Xếp được xây dựng bao gồm những chức năng sau:

• Về phần Dữ Liệu:

+ Khởi tạo số phần tử (tối đa 15 phần tử): người dùng có thể chọn chức năng tạo ngẫu nhiên hoặc tạo bằng tay do người dùng nhập vào.

+ Đặt lại dữ liệu mảng: người dùng có thể đặt lại dữ liệu của mảng thay đổi giá trị của phần tử trong mảng bằng chức năng ngẫu nhiên hoặc chọn vị trí phần tử cần thay đổi và điền giá trị của phần tử đó vào là xong.

•Về phần Điều Khiển:

+ Một dãy thuật toán bao gồm 8 thuật toán sắp xếp kinh điển cho người dùng có thể lựa chọn tùy ý

+ Có thể sắp xếp theo 2 hướng là sắp xếp tăng dần hoặc sắp xếp giảm dần

+ Trong quá trình chạy có thể điều chỉnh tốc độ sắp xếp của thuật toán trong khung chạy mô phỏng theo 3 tốc độ: Chậm – Trung Bình – Nhanh

+ Sau khi đã chọn những chức năng và thuật toán như ý, người dùng click vào Button Chạy để chương trình bắt đầu làm việc

+ Có thể Thoát chương trình bất cứ lúc nào bằng cách click vào Button Thoát

# **VII. KIỂM THỬ**

## **VII.1 Qui trình kiểm thử:**

• Kiểm thử đơn vị: Mỗi thành viên sau khi hoàn thành tính năng của mình sẽ tự kiểm thử để phát hiện tất cả các lỗi phát sinh và sau đó tự gỡ lỗi, nếu không fix được sẽ upload trên group facebook của nhóm để tất cả thành viên vào góp ý chỉnh sửa fix bug đó.

• Kiểm thử toàn bộ: Sau khi các tính năng hoàn chỉnh, nhóm trưởng sẽ tiến hành tổng hợp tất cả để hợp nhất các tính năng lại, các thành viên sẽ test tổng thể phần mềm ở các tính chất:

+ Tính đúng đắn: Kết quả trả về của phần mềm cho người dùng phải chính xác.

+ Tính tiện dụng và hiệu quả: Phần mềm phải cung cấp cho người dùng sự tiện lợi, thoải mái khi sử dụng.

+ Tính ổn định: Phát hiện tất cả các lỗi phát sinh cả về giao diện lẫn xử lý. Sau khi phát hiện lỗi, tiến hành thông báo để các thành viên khác sửa lỗi.

## **VII.2 Kết quả kiểm thử**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mức độ hoàn thành (%)** | **Ghi chú** |
| 1 | Khởi tạo số phần tử | 100% | Không xảy ra lỗi |
| 2 | Đặt Lại Dữ Liệu Mảng | 100% | Không xảy ra lỗi |
| 3 | Sắp Xếp Thuật Toán Theo Hướng Tăng Dần | 100% | Không xảy ra lỗi |
| 4 | Sắp Xếp Thuật Toán Theo Hướng Giảm Dần | 100% | Không xảy ra lỗi |
| 5 | Tốc Độ Sắp Xếp Thuật Toán “Nhanh” | 100% | Không xảy ra lỗi |
| 6 | Tốc Độ Sắp Xếp Thuật Toán “Trung Bình” | 100% | Không xảy ra lỗi |
| 7 | Tốc Độ Sắp Xếp Thuật Toán “Chậm” | 100% | Không xảy ra lỗi |
| 8 | Khung Chạy Mô Phỏng | 100% | Không xảy ra lỗi |

# **VIII. NHẬN XÉT VÀ KẾT LUẬN**

## **VIII.1. Đánh giá tổng quan**

• Nhìn chung phần mềm hoàn thành đầy đủ các chức năng cần thiết như:

* Chạy sắp xếp đúng kết quả
* Sắp xếp theo 2 hướng tăng dần hoặc giảm dần
* Màn hình mô phỏng trực quan, dễ nhìn
* Chương trình được thiết kế dễ sử dụng
* Cho phép nhập mảng ngẫu nhiên các phần tử nhanh gọn
* Tốc độ chạy của chương trình được thiết kế theo nhu cầu tùy ý của người dùng
* Mô tả chi tiết từng bước của thuật toán đang chạy tương ứng với từng dòng lệnh
* Có thể thay đổi giá trị của một phần tử ở vị trí bất kì

## **VIII.2 Ưu điểm và nhược điểm**

### ***VIII.2.1. Ưu điểm***

* Giao diện thân thiện dễ sử dụng với hầu hết mọi người dùng.
* Chức năng khá đầy đủ.
* Có thể sử dụng chương trình trong quá trình giảng dạy môn CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT về phần các thuật toán sắp xếp.
* Chương trình dùng được trên mọi nền tảng **IOS / WINDOW / LINUX**

### ***VIII.2.2. Nhược điểm***

* Input của chương trình là một mảng chỉ giới hạn 15 phần tử
* Chương trình chỉ có 8 thuật toán sắp xếp nên vẫn còn những thuật toán khác chưa được áp dụng vào.

### ***VIII.2.3. Hướng phát triển***

* Phát triển thêm các chức năng như sắp xếp nửa mảng đầu thì tăng theo thuật toán X còn nửa mảng sau thì giảm theo thuật toán Y
* Xây dựng thêm các thuật toán sắp xếp như Shaker Sort, Binary Insertion Sort, Radix Sort và những Sort khác đang được dùng nhiều hiện nay…

# **TÀI LIỆU THAM KHẢO**

1. <https://docs.oracle.com/javase/tutorial/essential/concurrency/procthread.html>

2. <http://www.tutorialspoint.com/java/java_thread_synchronization.htm>

3. <https://www.javatpoint.com/java-swing>

4. <https://docs.oracle.com/javase/tutorial/uiswing/components/index.html>

5. Slide môn học Cấu trúc dữ liệu và giải thuật