

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KÌ

**TRỰC QUAN HÓA THUẬT TOÁN SẮP XẾP
TRONG CẤU TRÚC DỮ LIỆU BẰNG MÔ HÌNH
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

Lớp học phần: OOPR230279_21_1_09

GVHD: TS. Lê Vĩnh Thịnh

Nhóm sinh viên thực hiện:

- | | |
|-------------------------|----------|
| 1. Nguyễn Kiều Châu Anh | 20110234 |
| 2. Lâm Hoàng Duyên | 20110174 |

Thành phố Hồ Chí Minh, tháng 11, năm 2021

LỜI CẢM ƠN

Để hoàn thành được đồ án, trước tiên, em xin được gửi lời cảm ơn chân thành đến Trường Đại học Sư phạm kỹ thuật đã tạo mọi điều kiện thuận lợi nhất cho quá trình học online của sinh viên khi dịch bệnh Covid 19 đang có nhiều diễn biến phức tạp. Tiếp đến, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn - Thầy Lê Vĩnh Thịnh - người không chỉ truyền đạt những kiến thức nền tảng của môn học cho chúng em trong quá trình học, mà còn hết sức hỗ trợ chúng em trong quá trình thực hiện đồ án. Trong thời gian được học với thầy, chúng em không những được xây dựng một nền tảng vững chắc cho môn học, mà còn được rèn luyện tinh thần tự học sao cho hiệu quả, nghiêm túc. Em tin rằng tất cả những kiến thức, trải nghiệm mà chúng em có được trong thời gian tham gia môn học của thầy đều sẽ là những hành trang quý báu để có cái nhìn đúng đắn hơn về ngành nghề mình đang theo.

Đồ án được nhóm chuẩn bị trong khoảng 2 tuần và chính thức làm trong hơn 6 tuần. Một quãng thời gian tuy không quá dài nhưng cũng không quá ngắn để chúng em có cơ hội củng cố lại kiến thức, rèn luyện kĩ năng lập trình, đọc hiểu code, tìm kiếm thông tin, fix bug, làm việc nhóm, viết báo cáo.... Mặc dù chúng em đã cố gắng hết sức, tuy nhiên, do kiến thức và kĩ năng còn nhiều hạn chế, đồ án khó tránh khỏi những thiếu sót. Chính vì vậy, chúng em rất mong nhận được những góp ý, nhận xét của thầy và các bạn để đồ án của nhóm được hoàn thiện hơn.

Một lần nữa, chúng em xin chân thành cảm ơn!

Họ và tên sinh viên: NGUYỄN KIỀU CHÂU ANH – MSSV: 20110234

Họ và tên sinh viên: LÂM HOÀNG DUYÊN – MSSV: 20110174

Ngành: Công nghệ thông tin

Tên đề tài: Trục quan hóa các thuật toán sắp xếp trong cấu trúc dữ liệu theo mô hình lập trình hướng đối tượng

Họ và tên Giáo viên hướng dẫn: TS. Lê Vĩnh Thịnh

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....
.....
.....

2. Ưu điểm:

.....
.....
.....

3. Khuyết điểm:

.....
.....
.....

4. Điểm: (Bằng chữ:)

.....

Tp. Hồ Chí Minh, ngày tháng năm 20...

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

MỤC LỤC

Chương 1: Tổng quan	6
1. Giới thiệu đề tài:.....	6
2. Mục tiêu:	7
3. Công cụ:	7
4. Phạm vi đề tài:.....	7
5. Phương pháp:	7
Chương 2: Cơ sở lí thuyết:.....	9
1. Lí thuyết về lập trình hướng đối tượng:.....	9
1.1. Tổng quan về lập trình hướng đối tượng:	9
1.2. Các tính chất của lập trình hướng đối tượng:	9
2. Lí thuyết về các thuật toán Sort được dùng trong đồ án:.....	10
2.1. Bubble Sort (Sắp xếp nổi bọt).....	10
2.2. Insertion Sort (Chèn trực tiếp):	11
2.3. Quick Sort (Sắp xếp nhanh):.....	12
3. Lí thuyết cơ bản về thiết kế giao diện và các chức năng bằng Winform:	14
3.1. Lí thuyết cơ bản về thiết kế giao diện:.....	14
3.2. Các chức năng của Winform:	14
4. Liên hệ giữa môn học lập trình hướng đối tượng và đồ án:	14
Chương 3: Triển khai đồ án:	16
3.1.Xác định yêu cầu:.....	16
3.2. Các chức năng chính của chương trình:	16

3.3. Thiết kế giao diện:	16
3.4. Xử lý yêu cầu người dùng:	19
3.4.1. Tổng quan về các lớp và đối tượng sử dụng:.....	19
3.4.2. Cách chương trình hoạt động:.....	21
Chương 4: Cài đặt và kiểm thử:.....	23
4.1. Môi trường lập trình:.....	23
4.3. Kiểm thử:	23
Chương 5: Kết luận:.....	25
1. Kế hoạch thực hiện:	25
2. Kết quả đạt được:	25
3. Khó khăn trong quá trình thực hiện	26
4. Kinh nghiệm rút ra:	26
TÀI LIỆU THAM KHẢO	27

Chương 1: Tổng quan

1. Giới thiệu đề tài:

Vào những năm 70 và 80, các hệ thống phần mềm chủ yếu được phát triển bởi các ngôn ngữ hướng thủ tục (procedural languages) như C, Pascal hay Fortran, trong đó, chương trình là một chuỗi các bước nối tiếp nhau tới khi hoàn thành. Nhưng khi các chương trình phải có kích thước lớn dần, lập trình hướng thủ tục là một rào cản khiến chương trình khó quản lý và bảo trì. Lập trình hướng cấu trúc (structured programming) được ra đời như một giải pháp cải tiến cho lập trình thủ tục (chia nhỏ chương trình thành các hàm con để gọi trong hàm main). Mặc dù so với lập trình thủ tục thì đây được xem là một bước tiến lớn, đối với những chương trình thực thi nhiều chức năng phức tạp và phải tương tác với nhiều hệ thống thì lập trình cấu trúc vẫn chưa thực sự tối ưu. Và như một điều tất yếu, phương pháp lập trình hướng đối tượng (Object Oriented Programming, viết tắt là OOP) được ra đời để giải quyết các vấn đề của lập trình cấu trúc. Lập trình hướng đối tượng đã chứng minh được tính ưu việt của mình so với hai người tiền nhiệm với hàng loạt các lợi ích như: dễ chuyển từ mô hình thực tế sang mô hình thực thi phần mềm, việc thay đổi chương trình nhanh chóng và hiệu quả hơn, có thể chia hệ thống thành nhiều phần nhỏ để chia cho các nhóm độc lập phát triển, có thể tạo giao diện người dùng trực quan,...

Nhóm thuật toán sắp xếp là một công cụ mạnh mẽ trong việc tiết kiệm thời gian sắp xếp các phần tử theo thứ tự, giảm dung lượng bộ nhớ, giúp việc tìm kiếm nhanh chóng và dễ dàng hơn. Trong cuộc sống, ứng dụng của các thuật toán sắp xếp có thể nói là rộng khắp. Danh sách học sinh thì cần phải sắp xếp theo thứ tự tên, điểm, lớp, trường,... quản lý số điện thoại khách hàng thì cần phải phân loại theo độ tiềm năng khách hàng, khu vực,... Vì vậy các thuật toán sắp xếp cũng được giảng dạy và nghiên cứu hết sức kỹ lưỡng trong chương trình học của sinh viên ngành công nghệ thông tin/ khoa học máy tính nói chung.

Từ hai cơ sở trên, một mặt để hiểu rõ hơn về cách hoạt động của mô hình hướng đối tượng, một mặt củng cố thêm kiến thức về các thuật toán sắp xếp, nhóm chúng em lựa chọn thực hiện đề án cuối kỳ vận dụng mô hình hướng đối tượng được trực quan hóa một số thuật toán sắp xếp.

2. Mục tiêu:

- Tìm hiểu thư viện vẽ giao diện động để xây dựng quá trình sắp xếp của các thuật toán sort.
- Phân tích, thiết kế và xây dựng các tính năng để người dùng có thể tùy chỉnh loại, mức độ cũng như chế độ sắp xếp.
- Tìm hiểu lý thuyết về các phương pháp của thuật toán sắp xếp.
- Triển khai các phương pháp đó vào winform.
- Vận dụng các kiến thức đã học để thực hiện Lập trình hướng đối tượng.

3. Công cụ:

- Mô hình hướng đối tượng bằng ngôn ngữ C#.
- Thiết kế giao diện bằng winform.

4. Phạm vi đề tài:

- Lập trình hướng đối tượng bằng C#.ul>- Sự tương tác giữa các lớp, các đối tượng khi thực hiện thuật toán Sort
- Cách tương tác giữa các đối tượng thuộc lớp thuật toán sort với giao diện winform.
- Các thuật toán Sort.
 - Cài đặt thuật toán theo mô hình hướng đối tượng 3 thuật toán Sort phổ biến gồm có: Bubble Sort, Insertion Sort, Quicksort.
- Winform
 - Thiết kế giao diện bằng Winform
 - Liên kết dữ liệu từ các lớp và đối tượng để trực quan hóa thuật toán sort trên màn hình winform.

5. Phương pháp:

- **Lý thuyết thực tiễn:** tìm hiểu tài liệu từ các website, tài liệu có liên quan đến đề án đang tìm hiểu.

- **Tham khảo chuyên gia:** Tham khảo và tiếp thu ý kiến của thầy hướng dẫn, của các đàn anh đi trước và bạn bè có kinh nghiệm liên quan đến các vấn đề của đề án.
- **Phát triển mô hình:** phát triển đề án thành phần mềm hướng đối tượng winform đơn giản. Trong đó, người dùng có thể tương tác thông qua các nhãn chọn, các nút lệnh, thấy và so sánh độ phức tạp của các thuật toán một cách trực quan.

Chương 2: Cơ sở lý thuyết:

1. Lý thuyết về lập trình hướng đối tượng:

- Lập trình hướng đối tượng (tiếng Anh: Object-oriented programming, viết tắt là OOP) là một mô hình lập trình dựa trên khái niệm "công nghệ đối tượng". Trong đó, đối tượng sẽ chứa đựng các dữ liệu, trên các trường, thường được gọi là các thuộc tính (có thể hiểu là biến); và mã nguồn, được tổ chức thành các phương thức (tương tự như hàm, thủ tục). Phương thức giúp cho đối tượng có thể truy xuất và hiệu chỉnh các trường dữ liệu của đối tượng khác, mà đối tượng hiện tại có tương tác. Các đối tượng trong các ngôn ngữ lập trình hướng đối tượng được xem như thực thể của một lớp, được dùng để định nghĩa một kiểu dữ liệu.

- Ưu điểm của OOP là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn.

- Các ngôn ngữ lập trình hướng đối tượng đáng chú ý gồm có: Java, C#, C++, Python, PHP, Ruby, Perl, Objective-C, Dart, Swift, Scala, Common Lisp, và Smalltalk. Trong phạm vi thực hiện của đồ án, nhóm sinh viên sẽ chủ yếu sử dụng ngôn ngữ C# để mô phỏng mô hình hướng đối tượng.

1.1. Tổng quan về lập trình hướng đối tượng:

1.2. Các tính chất của lập trình hướng đối tượng:

Lập trình hướng đối tượng có 4 tính chất chính sau:

1.2.1. Tính trừu tượng (abstraction):

Tính trừu tượng là nhìn nhận một đối tượng từ bên ngoài một cách tổng quát để tập trung vào cốt lõi và bỏ qua những chi tiết phức tạp, không phổ biến chung.

Tính trừu tượng thể hiện rất rõ ở lớp trừu tượng (abstract class), một lớp chứa một số đặc điểm chung cho nhiều đối tượng khác như là sự mở rộng nhưng bản thân chính nó không thể tự tạo ra các đối tượng.

1.2.2. Tính đóng gói (*encapsulation*):

Đây là tính chất đảm bảo sự toàn vẹn của đối tượng. Có thể hiểu đơn giản tính đóng gói như một tính chất giúp che giấu thông tin (information hiding) và chỉ cho phép hiển thị, tác động chỉnh sửa khi được cho phép.

Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào lập trình viên (thông qua việc cài đặt quyền truy cập là public, private, protected,...).

1.2.3. Tính đa hình (*polymorphism*):

Là tạo ra những cấu trúc cùng tên nhưng tại mỗi thời điểm có thể thực thi những nhiệm vụ khác nhau tùy thuộc vào dữ liệu truyền vào. Tức là, các đối tượng thuộc các lớp khác nhau có thể hiểu cùng một thông điệp cùng tên theo cách khác nhau mà không bị nhầm lẫn.

Ví dụ, khi định nghĩa hai đối tượng "hình_vuong" và "hình_tron" thì có một phương thức chung là "chu_vì". Khi gọi phương thức này thì nếu đối tượng là "hình_vuong" nó sẽ tính theo công thức khác với khi đối tượng là "hình_tron".

1.2.4. Tính kế thừa (*inheritance*):

Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa. Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

2. Lí thuyết về các thuật toán Sort được dùng trong đồ án:

2.1. Bubble Sort (Sắp xếp nổi bọt)

Ý tưởng: Xuất phát từ cuối dãy, đổi chỗ các cặp phần tử kề nhau khi phần tử trước lớn hơn phần tử sau.

Khi đó, sau khi kết thúc vòng lặp đầu, ta sẽ có phần tử bé nhất trong dãy hiện hành. Xem dãy hiện hành còn $n-1$ phần tử rồi lặp lại bước trên đến khi nào còn 1 phần tử.

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 $a[j] < a[j-1]$ thì Doicho($a[j], a[j-1]$);

$j = j-1$;

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

Nếu $i \geq N-1$: Hết dãy. Dừng

Ngược lại : Lặp lại Bước 2.

Cài đặt:

```
void BubbleSort(int a[], int n)
{
    int i, j;
    for (i = 0 ; i < n-2 ; i++)
        for (j = n-1; j > i ; j --)
            if (a[j] < a[j-1]) Swap(a[j], a[j-1]);
}
```

Độ phức tạp: $O(n^2)$.

2.2. Insertion Sort (Chèn trực tiếp):

Ý tưởng: giả sử có a_0, a_1, \dots, a_{N-1} đã có thứ tự, ta chèn phần tử thứ k vào đoạn sao cho có thứ tự, đảm bảo vị trí của a_k $a_{k-1} \leq a_k \leq a_{k+1}$

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 chỗ 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

Cài đặt

```

void InsertionSort(int d, int n )
{
    int pos, i;
    int x;//luu giá trị a[i] tránh bị ghi đè khi dời chỗ các
    phần tử.
    for(i=1 ; i<n ; i++) //đoạn a[0] đã sắp
    {
        x = a[i]; pos = i-1;
        // tìm vị trí chèn x
        while((pos >= 0)&&(a[pos] > x))
        { //kết hợp dời chỗ các phần tử sẽ đứng sau x trong dãy mới
            a[pos+1] = a[pos];
            pos--;
        }
        a[pos+1] = x; // chèn x vào dãy
    }
}

```

Độ phức tạp:

Tốt nhất: $O(n)$.

Xấu nhất: $O(n^2)$.

2.3. Quick Sort (Sắp xếp nhanh):

Ý tưởng

Giải thuật QuickSort sắp xếp dãy a_1, a_2, \dots, 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.

Sau khi thực hiện phân hoạch, dãy ban đầu được phân thành 3 đoạn:

- 1. $a_k \leq x$, với $k = 1 .. j$
- 2. $a_k = x$, với $k = j+1 .. i-1$
- 3. $a_k \geq x$, với $k = i..N$

Đoạn thứ 2 đã có thứ tự.

- Nếu các đoạn 1 và 3 chỉ có 1 phần tử: đã có thứ tự.

- Khi đó dãy con ban đầu đã được sắp.

- Nếu các đoạn 1 và 3 có nhiều hơn 1 phần tử thì dãy ban đầu chỉ có thứ tự khi các đoạn 1, 3 được sắp.

- Để sắp xếp các đoạn 1 và 3, ta lần lượt tiến hành việc phân hoạch từng dãy con theo cùng phương pháp phân hoạch dãy ban đầu vừa trình bày ...

```
void QuickSort(int a[], int left, int right)
{
    int i, j, x;
    x = a[(left+right)/2];
    i = left; j = right;
    while(i < j)
    {
        while(a[i] < x) i++;
        while(a[j] > x) j--;
        if(i <= j)
        {
            Doicho(a[i],a[j]);
            i++ ; j--;
        }
    }
    if(left<j) QuickSort(a, left, j);
    if(i<right) QuickSort(a, i, right);
}
```

```
}
```

Độ phức tạp:

- Tốt nhất $O(n \cdot \log n)$.
- Trung bình $O(n \cdot \log n)$.
- Xấu nhất $O(n^2)$.

3. Lí thuyết cơ bản về thiết kế giao diện và các chức năng bằng Winform:**3.1. Lí thuyết cơ bản về thiết kế giao diện:**

- Quen thuộc với người dùng.
- Thống nhất: hệ thống nên hiển thị ở mức thống nhất thích hợp. Ví dụ: các câu lệnh và menu nên có cùng định dạng...
- Tương tự: nếu một yêu cầu được xử lý theo cách thông thường thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tự.
- Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi tới tình trạng trước đó: undo, xác nhận một lần nữa trước khi sửa xóa...
- Có tính hướng dẫn.
- Tính đa dạng: hỗ trợ nhiều loại tương tác cho nhiều loại người sử dụng khác nhau. Ví dụ: hiển thị phông chữ lớn với những người cận thị.

3.2. Các chức năng của Winform:

- Tạo các button, text box, label, Groupbox, RadioButton, CheckBox, ComboBox,...
- Gắn các event cho các Button.
- Khi viết code: có thể lấy text từ TextBox cho tới show dữ liệu bằng MessageBox hoặc dùng Grid để kết nối SQL.

4. Liên hệ giữa môn học lập trình hướng đối tượng và đồ án:

Cơ sở lí thuyết của lập trình hướng đối tượng được vận dụng chủ yếu qua:

- Làm việc với các class và đối tượng (chủ yếu là từ class algorithm và class frmMain)

- Đảm bảo tính đóng gói trong lập trình hướng đối tượng cho các biến có từ khóa truy cập là `private`, tạo các `properties` để truy cập đến các biến thành viên.
- Xây dựng các `constructor` để khởi tạo giá trị hợp lệ ban đầu cho các đối tượng của `class`.

Chương 3: Triển khai đề án:

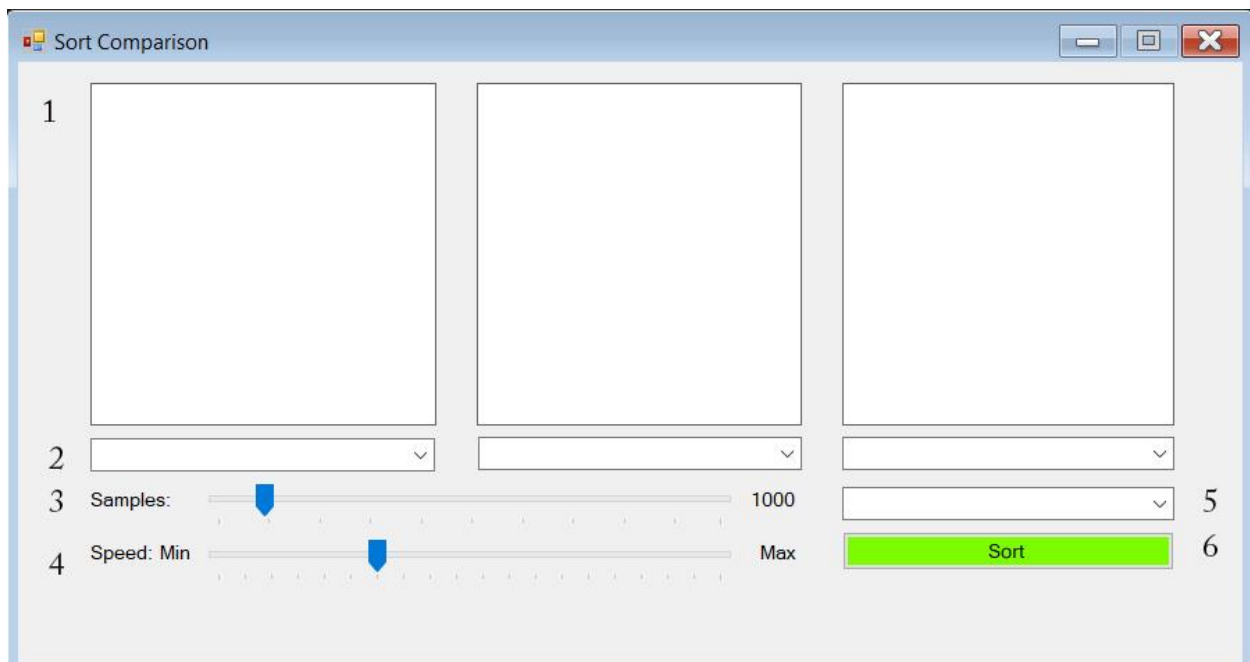
3.1.Xác định yêu cầu:

- Xây dựng một phần mềm winform có giao diện đơn giản, dễ sử dụng, thực hiện việc sắp xếp và trực quan hóa một số thuật toán sắp xếp.
- Người dùng có thể tương tác với chương trình thông qua các nút chọn.

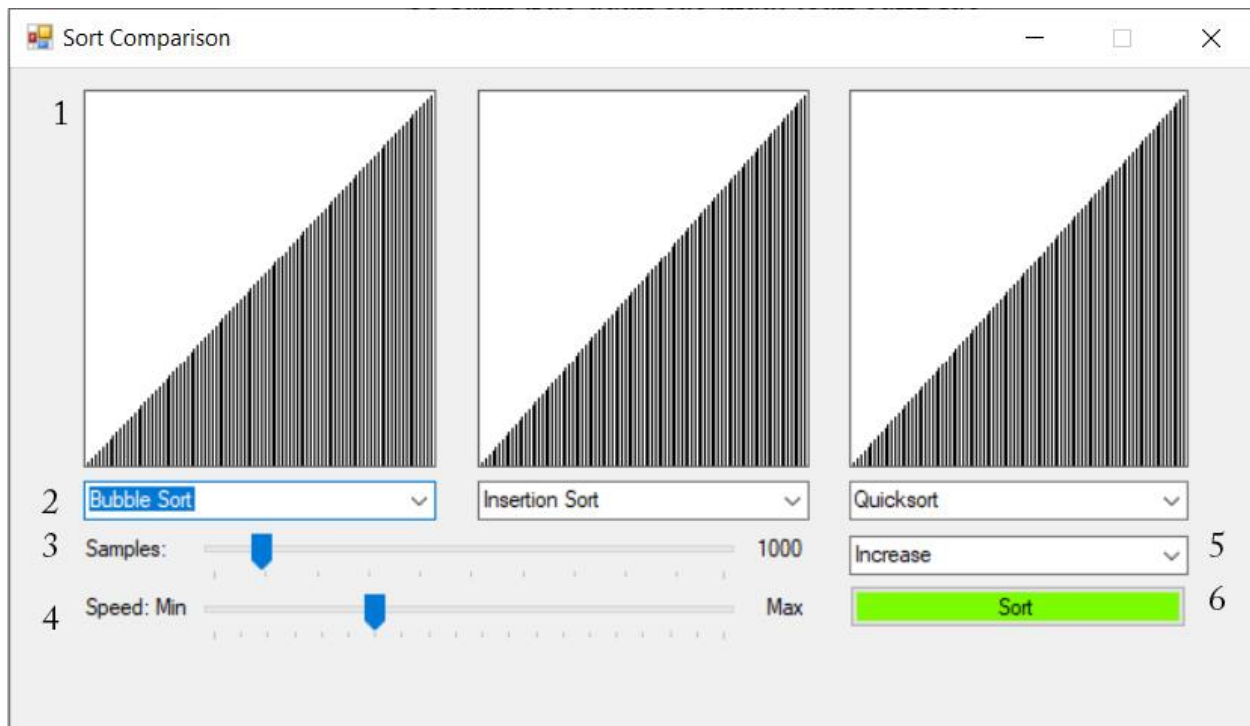
3.2. Các chức năng chính của chương trình:

- Chọn lựa thuật toán sắp xếp.
- Chọn sắp xếp tăng hay giảm.
- Chọn số phần tử sắp xếp.
- Điều chỉnh tốc độ sắp xếp nhanh chậm.
- So sánh trực quan các thuật toán cùng lúc.

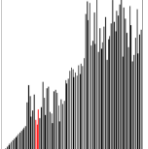
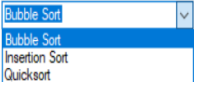
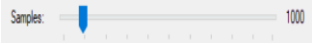

3.3. Thiết kế giao diện:





Hình 3.3.1. Giao diện trước khi thực thi



Hình 3.3.2. Giao diện sau khi thực thi

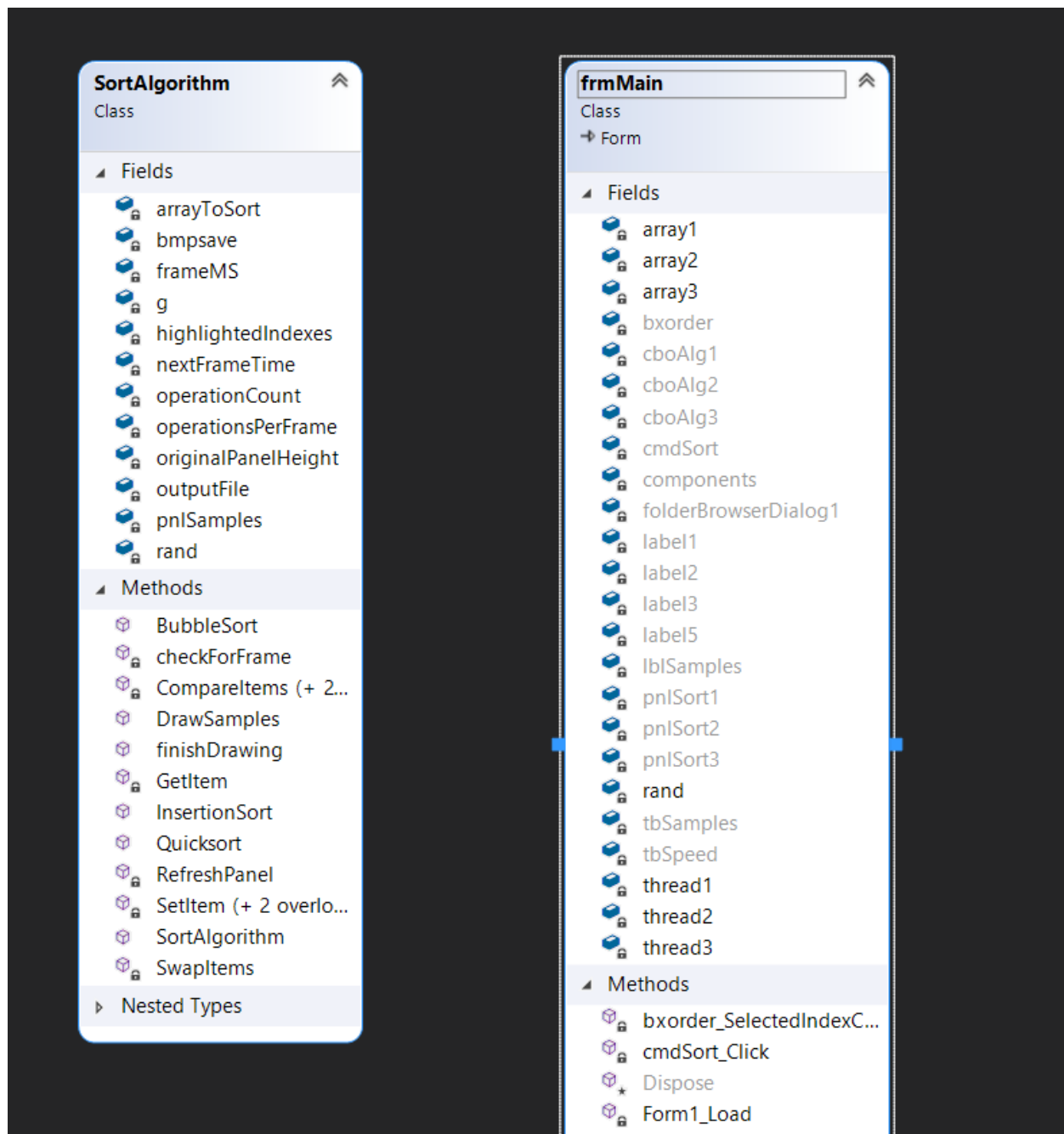
STT	Giao diện	Chức năng
1		Display quá trình thực hiện của các thuật toán sort khi thực thi chương trình.
2		Đưa ra các lựa chọn về loại thuật toán trước khi thực thi chương trình.
3		Người dùng di chuyển vạch xanh trên thanh kéo để điều chỉnh số lượng phần tử sắp xếp trước khi thực thi chương trình (tối thiểu 10 phần tử, tối đa 1000 phần tử).
4		Người dùng di chuyển vạch xanh trên thanh kéo để điều chỉnh tốc độ sắp xếp trước khi thực thi chương trình.

5		Đưa ra các lựa chọn về sắp xếp tăng/ giảm trước khi thực thi chương trình.
6		Khi nhấn nút sort chương trình sẽ minh họa quá trình thực thi và hiển thị kết quả sắp xếp dưới dạng biểu đồ trực quan (hình 3.3.2)

Bảng 3.3.3. Bảng chức năng

3.4. Xử lý yêu cầu người dùng:

3.4.1. Tổng quan về các lớp và đối tượng sử dụng:



Hình 3.3.5. Class diagram của chương trình

STT	Tên lớp	Thuộc tính của lớp	Phương thức của lớp	Đối tượng được tạo ra từ lớp	Chức năng
1	SortAlgorithm	<ul style="list-style-type: none"> - arrayToSort - bmpsave - frameMS - g - highlightedIndexes - nextFrameTime - operationCount - operationsPerFrame - originalPanelHeight - pnlSamples - rand 	<ul style="list-style-type: none"> - BubbleSort() - checkForFrame() - CompareItems(IList arrayToSort, int index1, int index2) - CompareItems(IList arrayToSort, int index1, object o) - CompareItems(IList arrayToSort, object o, int index1) - DrawSamples() - finishDrawing() - GetItem() - InsertionSort() - Quicksort() - RefreshPanel() - SetItem(IList arrayToSort, int toIndex, int fromIndex) 		Xây dựng các thuật toán sort và mô tả cách sắp xếp của các thuật toán bằng hình ảnh cụ thể

			- SetItem(IList arrayToSort, int toIndex, object fromObject) - SetItem(IList arrayToSort, ref object toObject, int fromIndex) - SortAlgorithm() - SwapItems()		
2	frmMain	- array1 - array2 - array3 - rand - thread1 - thread2 - thread3	- cmdSort_Click() - Form1_Load() - frmMain() -frmMain_FormClosing() - isSorted() - PrepareForSort() - Randomize()	- sa - sa2 - sa3	Tạo ra các array được random và nhận yêu cầu của người dùng để truy cập tới các hàm trong lớp SortAlgorith m thực hiện chương trình

Bảng 3.3.4. Bảng tóm tắt các class, object

3.4.2. Cách chương trình hoạt động:

- Đầu tiên, người dùng thiết lập các yêu cầu (loại thuật toán, số phần tử, tốc độ sắp xếp, sắp xếp tăng/ giảm) thông qua các nút chọn.
- Sau khi nhận lệnh từ người dùng, chương trình sẽ truy cập tới class frmMain (gọi về chức năng tương ứng...).

+ Đầu tiên chương trình sẽ lấy các yêu cầu của người dùng từ frmMain[Design] và tạo ra 3 array tương ứng với số lượng phần tử mà người dùng yêu cầu sắp xếp cho 3 khung chạy.

+ Sau đó chương trình sẽ tạo ra 3 đối tượng (sa, sa2, sa3) kiểu SortAlgorithm có kèm theo tốc độ sắp xếp và xét xem người dùng yêu cầu thuật toán sort nào để truy cập đến hàm xây dựng thuật toán sort tương ứng. Song song bên cạnh là dùng hàm checkForFrame() để vẽ quá trình sắp xếp vào khung cho sẵn trong class SortAlgorithm.

Chương 4: Cài đặt và kiểm thử:

4.1. Môi trường lập trình:

- Hệ điều hành Window 10 (10.0 Build 19043).

-IDE: Visual studio 2019.

4.3. Kiểm thử:

Nhóm đã tiến hành kiểm thử các chức năng chính, tuy nhiên do thời gian dành cho việc kiểm thử chưa nhiều nên số lượng test case còn hạn chế. Sau khi thực hiện xong việc kiểm thử thì nhóm xin trình bày một số test case tiêu biểu.

STT	Mô tả test case	Test steps	Kết quả dự kiến	Trạng thái
1	Kiểm tra việc lựa chọn thuật toán.	1. Bấm nút mũi tên để hiển thị các lựa chọn. 2. Chọn thuật toán.	Thuật toán được chọn sẽ hiện ra và trực quan hóa khi bấm nút Sort. VD: Sau khi người dùng chọn “Insertion Sort”, dòng “Insertion Sort” sẽ hiện lên ở thanh chọn.	PASS
2	Kiểm tra việc lựa chọn số phần tử sắp xếp.	1. Kéo thanh trượt Samples (trong phạm vi từ 10 tới 1000). 2. Dùng kéo thanh trượt khi đạt được số phần tử như mong muốn.	Thanh trượt di chuyển tới đâu thì số phần tử tương ứng sẽ hiện phía ngoài cùng bên trái thước kéo (cạnh chữ “Samples”). Sau khi bấm nút Sort, thuật toán sẽ được sắp xếp với số phần tử mà người dùng đã chọn..	PASS

3	Kiểm tra việc lựa chọn tốc độ sắp xếp.	<p>1. Kéo thanh trượt Speed (từ Min tới Max).</p> <p>2. Dừng kéo thanh trượt khi đạt được số tốc độ sắp xếp như mong muốn.</p>	Người dùng có thể kéo thanh trượt để điều chỉnh tốc độ. Tốc độ nhanh chậm có thể nhận thấy một cách trực quan khi thuật toán chạy.	PASS
4	Kiểm tra việc lựa chọn tăng giảm.	<p>1. Bấm nút mũi tên để hiện thị các lựa chọn.</p> <p>2. Chọn Increase (tăng) hoặc Decrease (giảm).</p>	Trong quá trình thực thi, thuật toán sẽ thực hiện sắp xếp tăng/ giảm.	PASS
5	Kiểm tra quá trình thực thi các thuật toán sau khi bấm nút Sort.	<p>1. Bấm nút Sort sau khi thiết lập xong các lựa chọn.</p> <p>2. Quan sát chương trình chạy trên màn hình winform.</p>	<p>Các thuật toán được thực thi và thể hiện trực quan trong 3 khung biểu diễn trên màn hình.</p> <p>Giao diện không bị vỡ, các nút, nhãn,.. không bị đè lên nhau.</p> <p>Định dạng chữ, màu sắc không bị thay đổi.</p>	PASS

Bảng 4.1. Bảng test cases

Chương 5: Kết luận:

1. Kế hoạch thực hiện:

STT	Thời gian	Công việc
1	21/09	-Nhận đồ án, lập nhóm
2	22/09-10/10	-Tham khảo báo cáo của các khóa trước. -Hệ thống lại kiến thức về OOP. -Tìm hiểu các thuật toán sắp xếp.
3	11/10-18/10	-Triển khai thuật toán sắp xếp theo mô hình hướng đối tượng. -Tham khảo các dự án mã nguồn mở tương tự trên các trang chia sẻ mã nguồn mở.
4	19/10-22-10	-Phân tích dự án tham khảo và thống nhất hướng phát triển thêm một số chức năng.
5	23/10-14/11	-Code các chức năng thêm vào: Tăng lên 3 khung biểu diễn, thêm chức năng chọn sắp xếp tăng/ giảm, điều chỉnh lại thiết kế giao diện winform.
6	15/11-18/11	-Kiểm thử các chức năng chính.
7	19/11	-Lập sùron bài báo cáo.
8	20/11-1/12	-Viết và hoàn thiện báo cáo đồ án.
9	7/12	-Báo cáo đồ án.

Bảng 5.1. Bảng kế hoạch

2. Kết quả đạt được:

- Hiểu rõ kiến thức nền tảng về lập trình hướng đối tượng.
- Củng cố kiến thức về các thuật toán sắp xếp được học trong môn Cấu trúc dữ liệu và giải

thuật.

-Nắm được cách triển khai mô hình hướng đối tượng trong một dự án winform cụ thể.

3. Khó khăn trong quá trình thực hiện

STT	Khó khăn	Khắc phục
1	Kiến thức về winform chỉ ở mức cơ bản nên còn nhiều hạn chế, khi gặp bug bị lúng túng.	Tìm cách fix bug trên các diễn đàn trao đổi như stack overflow.
2	Vì dịch bệnh nên không có thời gian gặp gỡ, trao đổi.	Họp mặt qua các nền tảng trực tuyến như google meet.

4. Kinh nghiệm rút ra:

Sau quá trình thực hiện đồ án, bản thân từng thành viên trong nhóm đã có thêm nhiều kinh nghiệm để sau này có thể làm việc trong những dự án khác, trong các môi trường làm việc khác một cách tốt hơn.

- Cần có một kế hoạch cụ thể, phân chia rõ ràng từ đầu, không để xảy ra tình trạng mất phương hướng, thiếu sự phối hợp trong thời gian đầu.
- Luôn back up nhiều bản đồ án phòng khi có vấn đề xảy ra thì không phải làm lại từ đầu.

TÀI LIỆU THAM KHẢO

[1] Winform là gì, <https://plctech.com.vn/winform-la-gi/>, truy cập ngày 19/11/2021.

[2] Lập trình hướng đối tượng,

https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng, truy cập ngày 15/11/2021.

[3] Visualization and comparison of sorting algorithm in C#,

https://www.codeproject.com/Articles/1087568/Visualization-and-Comparison-of-sorting-algorith?fbclid=IwAR0Zf0X9kA9uo2BmgeXbusYa1qaT8_cCkveamxpEgIzWgYCBMraGn2_XzUk, truy cập ngày 11/10/2021