BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, icon

Description automatically generated

**BÁO CÁO THỰC TẬP CƠ SỞ**

**MÔ PHỎNG THUẬT TOÁN SẮP XẾP CHÈN BẰNG C#**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Hoàng Quốc Nam**

**Mã số sinh viên: 61133985**

Khánh Hòa - 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, icon

Description automatically generated

**BÁO CÁO THỰC TẬP CƠ SỞ**

**MÔ PHỎNG THUẬT TOÁN SẮP XẾP CHÈN BẰNG C#**

GVHD: ThS. Đoàn Vũ Thịnh

SVTH: Hoàng Quốc Nam

MSSV: 61133985

Khánh Hòa, Tháng 1/2022

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa: Công nghệ Thông tin**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ**

**Tên đề tài:** MÔ PHỎNG THUẬT TOÁN SẮP XẾP CHÈN BẰNG C#

**Giảng viên hướng dẫn:** ThS. Đoàn Vũ Thịnh

**Sinh viên được hướng dẫn:** Hoàng Quốc Nam

**MSSV:** 61133820

**Khóa:** 61 **Ngành:** Công nghệ Thông tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 | 2/12/2021 | Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. Sinh viên trình bày kế hoạch thực hiện. | Sinh viên và GVHD trao đổi nội dung của đề tài. Phân chia công việc theo từng thời gian sao cho phù hợp với yêu cầu. |
| 2 | 6/12/2021 | Sinh viên trình bày việc mô phỏng thuật toán chính dựa trên kiến thức đã được học ở môn kỹ thuật đồ họa và các kiến thức thu nhận được từ Internet để minh họa bài toán đa dạng nhất có thể. | Sinh viên hiểu được vấn đề cần phải thực hiện và có giải pháp cho từng vấn đề cụ thể. Tuy nhiên một vấn đề hoàn toán mới là sử dụng thư viện của chuột hay làm giao diện thì chưa làm lần nào nên cần đầu tư nhiều thời gian hơn. |
| 3 | 14/12/2021 | Sinh viên hoàn thiện các thuật toán đã đề ra với dữ liệu đầu vào được nhập từ bàn phím. Trình bày thuật toán với các trường hợp sai và chỉ ra được hướng khắc phục cho các trường hợp đó. | Sinh viên hiểu nội dung của thuật toán khá chi tiết với các trường hợp hạn chế của thuật toán. Việc lập trình cũng hoàn thành ở mức độ nhập dữ liệu từ bàn phím bằng giao diện nhưng vấn đề với chuột thì chưa thực hiện được. |
| 4 | 21/12/2021 | Sinh viên trình bày bản thảo của báo cáo thực tập lần thứ 1 và tiến hành chỉnh sửa. | Báo cáo chỉ trình bày chung chung chưa đi vào cụ thể phân tích các yêu cầu của bài toán, hình ảnh, bảng biểu chưa trình bày rõ ràng. Cần hiệu chỉnh theo yêu cầu của GVHD. |
| 5 | 28/01/2021 | Sinh viên trình bày bản thảo lần 2 | Báo cáo lần này đã khắc phục được các lỗi của lần trước, tuy nhiên phần phương pháp và kết quả chưa nổi bât, chưa có sự liên kết giữa các phần. |
| 6 | 4/1/2022 | Sinh viên nộp bản thảo lần cuối sau khi đã chỉnh sửa các yêu cầu như đã đề ra. | Sinh viên nghiêm túc chỉnh sửa báo cáo theo định hướng của GVHD. |

**Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL)**:

Sinh viên thực hiện tốt các yêu cầu của GVHD, trong quá trình thực hiện đề tài có sự liên hệ chặt chẽ với GV. Theo lịch hẹn Sinh viên đều có mặt để trình bày ý tưởng của các nội dung lần trước. Trong quá trình hoàn tất báo cáo đều nỗ lực không ngừng mặc dù đang cao điểm của đợt thi học kỳ nhưng SV vẫn dành thời gian không ít cho TTCS.

Về nội dung báo cáo đã thỏa mãn các yêu cầu của đề tài như trong đề cương. Về kết quả chương trình đã minh họa được thuật toán. Về các yêu cầu cao hơn như sử dụng thư viện của chuột hay minh họa các trường hợp nhược điểm của thuật toán thì chưa thực hiện thành công.

Về hình thức của báo cáo và sản phẩm, báo cáo trình bày rõ ràng các mục tiêu, phương pháp, kết quả và thảo luận cho sản phẩm. Còn về sản phẩm như đã trình bày có phần hạn chế.

Điểm hình thức: …./10 Điểm nội dung: …../10 **Điểm tổng kết: …../10**Đồng ý cho sinh viên: Được bảo vệ:  Không được bảo vệ: 

Khánh Hòa, ngày 20 tháng 01 năm 2021

Cán bộ hướng dẫn

*(Ký và ghi rõ họ tên)*

# **LỜI CẢM ƠN**

Để có thể hoàn thành đợt thực tập lần này, em xin chân thành cảm ơn đến quý thầy cô khoa Công nghệ Thông tin đã tạo điều kiện hỗ trợ và giúp đỡ em trong quá trính học tập và nghiên cứu đề tài này.

Qua đây, em xin chân thành cảm ơn thầy Đoàn Vũ Thịnh, người đã trực tiếp quan tâm và hướng dẫn chúng em hoàn thành tốt đợt thực tập trong thời gian qua.

Do kiến thức còn hạn chế và thời gian thực hiện còn ngắn nên bài báo cáo của em còn nhiều thiếu sót, kính mong sự góp ý của quý thầy cô.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI CẢM ƠN** 3](#_Toc92233595)

[**TÓM TẮT** 6](#_Toc92233596)

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 6](#_Toc92233597)

[**1.1** **Sắp xếp chèn là gì?** 6](#_Toc92233598)

[**1.2** **Ý tưởng thuật toán sắp xếp chèn** 6](#_Toc92233599)

[**1.3** **Ngôn ngữ C#** 6](#_Toc92233600)

[**1.4** **Winform là gì?** 7](#_Toc92233601)

[**1.4.1** **Ưu điểm các phần mềm trên Winform**: 7](#_Toc92233602)

[**1.4.2** **Nhược điểm các phần mềm trên Winform:** 7](#_Toc92233605)

[**1.5** **Các thư viện thường được sử dụng** 8](#_Toc92233606)

[**Chương 2. PHƯƠNG PHÁP NGHIÊN CỨU** 9](#_Toc92233611)

[**2.1** **Tổng quan về giao diện mô phỏng** 9](#_Toc92233612)

[**2.2** **Phân tích thuật toán** 12](#_Toc92233613)

[**2.2.1** **Chức năng nút tạo mảng** 13](#_Toc92233614)

[**2.2.2** **Tạo mảng:** 13](#_Toc92233615)

[**2.2.3** **Tạo mảng kết quả** 16](#_Toc92233616)

[**2.2.4** **Xóa mảng** 17](#_Toc92233617)

[**2.2.5.1** **Node qua phải** 19](#_Toc92233618)

[**2.2.5.2** **Node sang trái** 19](#_Toc92233619)

[**2.2.5.3** **Node đi lên** 20](#_Toc92233620)

[**2.2.5.4** **Node đi xuống** 20](#_Toc92233621)

[**2.2.6** **Các hàm khác** 20](#_Toc92233622)

[**2.2.6.1** **Hàm ngủ** 20](#_Toc92233623)

[**2.2.6.2** **Hàm trễ** 21](#_Toc92233624)

[**2.2.6.3** **Hàm hoán đổi vị trí 2 Node** 21](#_Toc92233625)

[**2.2.6.4** **Hàm hoàn thành** 22](#_Toc92233626)

[**2.2.7** **Nhập dữ liệu** 22](#_Toc92233627)

[**2.2.8** **Hàm sắp xếp** 23](#_Toc92233628)

[**2.2.8.1** **Sắp xếp tăng dần** 23](#_Toc92233629)

[**2.2.8.2** **Sắp xếp giảm dần** 25](#_Toc92233630)

[**2.2.9** **Chức năng nút sắp xếp** 26](#_Toc92233631)

[**CHƯƠNG 3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỀN** 28](#_Toc92233632)

[3.1 Kết luận 28](#_Toc92233633)

[3.2 Hướng phát triển 28](#_Toc92233634)

[**TÀI LIỆU THAM KHẢO** 29](#_Toc92233635)

# **DANH MỤC HÌNH**

[Hình 2.1.1. Giao diện chính lúc mới mở 11](#_Toc92279915)

[Hình 2.1.2. Giao diện sau khi tạo mảng 12](#_Toc92279916)

[Hình 2.1.3. Giao diện nhập liệu 12](#_Toc92279917)

[Hình 2.1.4. Giao diện sau khi nhập giá trị 13](#_Toc92279918)

[Hình 2.1.5. Các node trong khi sắp xếp 13](#_Toc92279918)

[Hình 2.1.6. Giao diện sau khi sắp xếp. 14](#_Toc92279918)

[Hình 2.2.5. Thông báo tạo sai mảng 16](#_Toc92279919)

[Hình 2.2.6. Thiết lập thuộc tính cho node 16](https://d.docs.live.net/63fa808edf50a047/Documents/Báo%20cáo%20TTCS.docx#_Toc92279920)

[Hình 2.2.7. Tạo các mảng dữ liệu 16](#_Toc92279921)

[Hình 2.2.8. Tạo node 17](#_Toc92279922)

[Hình 2.2.9. Tạo chỉ số 17](#_Toc92279923)

[Hình 2.2.10. Khai báo hàm tạo mảng kết quả 18](#_Toc92279924)

[Hình 2.2.11. Tạo node mảng kết quả 18](#_Toc92279925)

[Hình 2.2.12. Tạo chỉ số mảng kết quả 19](#_Toc92279926)

[Hình 2.2.13. Xóa mảng 20](#_Toc92279927)

[Hình 2.2.14. Node qua phải 20](#_Toc92279928)

[Hình 2.2.15. Node qua trái 21](#_Toc92279929)

[Hình 2.2.16. Node đi lên 22](#_Toc92279930)

[Hình 2.2.17. Node đi xuống 22](#_Toc92279931)

[Hình 2.2.18. Hàm ngủ 23](#_Toc92279932)

[Hình 2.2.19. Hàm trễ 23](#_Toc92279933)

[Hình 2.2.20. Hàm hoán vị node 23](#_Toc92279934)

[Hình 2.2.21. Hàm hoàn thành 24](https://d.docs.live.net/63fa808edf50a047/Documents/Báo%20cáo%20TTCS.docx#_Toc92279935)

[Hình 2.2.22. Chức năng nút nhập dữ liệu 24](#_Toc92279936)

[Hình 2.2.23. Chức năng nút sắp xếp 28](#_Toc92279937)

# **TÓM TẮT**

Hiện nay trong hầu hết các hệ lưu trữ, quản lý dữ liệu, thao tác tìm kiếm thường được thực hiện nhiều nhất để khai thác thông tin một cách nhanh chóng và muốn việc tìm kiếm nhanh chóng thì dữ liệu cần phải được sắp xếp sẵn, ngăn nắp theo thứ tự, hệ thống nhất định. Vì vậy sắp xếp có ý nghĩa nhất định trong việc quản lý và lưu trữ thông tin. Nội dung giới thiệu trình bày dưới đây là thuật toán sắp xếp thông dụng mà tôi nghiên cứu và mô phỏng: Thuật toán sắp xếp chèn (Insertion Sort). Sắp xếp chèn là một thuật toán sắp xếp bắt chước cách sắp xếp quân bài của những người chơi bài. Muốn sắp xếp một bộ bài theo trật tự người chơi rút lần lượt từ quân thứ 2, so với các quân đứng trước nó để chèn vào vị trí thích hợp. Quy trình thực hiện mô phỏng được trải qua các bước từ thiết kế giao diện, hiển thị kết quả đầu ra trên môi trường WinForms (C#) qua ứng dụng Visual Studio. Kết quả của việc cài đặt thuật toán đáp ứng với các yêu cầu đặt ra của đợt thực tập cơ sở này.

# **CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

* 1. **Sắp xếp chèn là gì?**

Sắp xếp chèn là một thuật toán sắp xếp bắt chước cách sắp xếp quân bài của những người chơi bài. Muốn sắp xếp một bộ bài theo trật tự người chơi rút lần lượt từ quân thứ 2, so với các quân đứng trước nó để chèn vào vị trí thích hợp.

* 1. **Ý tưởng thuật toán sắp xếp chèn**

Thuật toán sắp xếp chèn thực hiện sắp xếp dãy số theo cách duyệt từng phần tử và chèn từng phần tử đó vào đúng vị trí trong mảng con(dãy số từ đầu đến phần tử phía trước nó) đã sắp xếp sao cho dãy số trong mảng sắp đã xếp đó vẫn đảm bảo tính chất của một dãy số tăng dần.

Khởi tạo mảng với dãy con đã sắp xếp có k = 1 phần tử(phần tử đầu tiên, phần tử có chỉ số 0)

Duyệt từng phần tử từ phần tử thứ 2, tại mỗi lần duyệt phần tử ở chỉ số i thì đặt phần tử đó vào một vị trí nào đó trong đoạn từ [0…i] sao cho dãy số từ [0…i] vẫn đảm bảo tính chất dãy số tăng dần. Sau mỗi lần duyệt, số phần tử đã được sắp xếp k trong mảng tăng thêm 1 phần tử.

Lặp cho tới khi duyệt hết tất cả các phần tử của mảng.

Ví dụ minh họa:

A picture containing text, electronics, calculator, keyboard

Description automatically generated

Hình 1.1. Minh họa sắp xếp chèn

* 2. **Ngôn ngữ C#**

C Sharp (C#) là [ngôn ngữ lập trình](https://vietnix.vn/ngon-ngu-lap-trinh/) hướng đối tượng mạnh mẽ và đa năng. Được phát triển bởi Microsoft vào năm 2000. Microsoft phát triển C# dựa trên C++ và Java và C Sharp cũng được xem là ngôn ngữ cân bằng giữa C++, Visual Basic, Delphi và Java.

C# hỗ trợ việc tạo ra Windows Forms dựa trên thuật ngữ Winform

* 1. **Winform là gì?**

Winform là thuật ngữ mô tả một ứng dụng được viết dùng .NET FrameWorrk và có giao diện người dùng Windows Forms.

Mỗi màn hình windows cung cấp một giao diện giúp người dùng giao tiếp với ứng dụng. Giao diện này được gọi là giao diện đồ họa (GUI) của ứng dụng.

* + 1. **Ưu điểm các phần mềm trên Winform**:

Đa phần lập trình viên C#. NET nào cũng từng học/sử dụng Winform. Bởi vì: Giao diện kéo thả dễ sử dụng; Gắn các event cho các button chỉ cần double click, lại hỗ trợ quá trời event như click, hover,…; Việc viết code cũng vô cùng trực quan: từ việc lấy text từ TextBox cho tới show dữ liệu bằng MessageBox, hoặc dùng Grid để kết nối SQL. **WinForm rất dễ học và dễ dạy**.

Vì dễ code, chỉ cần kéo thả, lại có nhiều component có sẵn, WinForm rất phù hợp để làm các phần mềm quản lý, tính tiền, thống kê… . Đây cũng là loại ứng dụng mà các công ty/doanh nghiệp vừa và nhỏ cần.

* + 1. **Nhược điểm các phần mềm trên Winform:**

Phần mềm chạy trên nền tảng Windows đó chính là người dùng muốn dùng phần mềm sẽ phải sử dụng máy tính đã cài phần mềm. Do vậy, bạn phải mang theo mình chiếc máy tính cá nhân để phục vụ cho công việc.

Winform chỉ phù hợp các ứng dụng trên desktop: ứng dụng quản lý thông tin, ứng dụng tương tác trực tiếp với người dùng.

Đồ họa trên winform không cao nên giao diện phần mềm sẽ thiếu tính trực quan, hơi khó thao tác, không thân thiện với người dùng.

* 1. **Các thư viện thường được sử dụng**

**System.Drawing:** Cung cấp quyền truy cập vào chức năng đồ họa cơ bản của GDI +.

**System.Window.Forms:** Chứa các lớp để tạo các ứng dụng dựa trên Windows tận dụng tối đa các tính năng giao diện người dùng phong phú có sẵn trong hệ điều hành Microsoft Windows.

**DevExpress.XtraEditors:** Chứa các lớp biên tập viên. Chúng cung cấp chức năng cho các điều khiển tương ứng.

**System.Threading:** Cung cấp các lớp và giao diện cho phép lập trình đa luồng.

# **Chương 2. PHƯƠNG PHÁP NGHIÊN CỨU**

1. **Tổng quan về giao diện mô phỏng**

Giao diện mô phỏng sẽ được tạo ra dựa trên 2 giao diện.

Giao diện chính sẽ được dùng để mô tả các thông tin của đề tài như: tên đề tài, tên giáo viên hướng dẫn, tên sinh viên thực tập, mảng trước khi được sắp xếp, mảng sau khi sắp xếp. Các GroupBox chứa các Button dùng để tạo mảng, nhập số liệu mảng, chọn cách sắp xếp tang hay giảm dần và nút bắt đầu tiến trình sắp xếp.

Giao diện nhập liệusẽ được dùng để nhập giá trị cho các phần tử trong mảng được sắp xếp.

**Graphical user interface, text, application

Description automatically generated**

Hình 2.1.1. Giao diện chính lúc mới mở

Như Hình 2.1.1, ban đầu giao diện chính sẽ gồm có 3 label chứa thông tin của tên đề tài, giáo viên hướng dẫn, sinh viên thực tập bao gồm: “Mô Phỏng Giải Thuật Sắp Xếp”, “GVHD: Đoàn Vũ Thịnh”, “SVTT Hoàng Quốc Nam”.

Các GroupBox như “Dữ liệu” dùng để lựa chọn số lượng phần tử cho mảng cần sắp xếp với số lượng của mảng lớn hơn hoặc bằng 2 và bé hơn hoặc bằng 15, cùng với đó là Button “Tạo Mảng” để mở ra Giao diện nhập liệu.

Sau khi hoàn tất tạo mảng giao diện chính sẽ được trình bày như hình 1.2:

**A picture containing diagram

Description automatically generated**

Hình 2.1.2. Giao diện sau khi tạo mảng

Mảng được tạo ra bởi các phần tử, các phần tử được tạo từ các node và được đánh số từ node số 0 đến node số n-1.

Lúc này GroupBox “Tạo giá trị cho mảng” sẽ được phép sử dụng. Button “Nhập” được sử dụng để nhập dữ liệu cho các node. Sau khi nhấn vào “Button” nhập.

Giao diện nhập liệu sẽ xuất hiện.

**Graphical user interface

Description automatically generated**

Hình 2.1.3. Giao diện nhập liệu

Giao diện nhập liệu bao gồm 1 label “Phần tử thứ”, 2 TextBox dùng để hiển thị vị trí của node và giá trị của node. Sau mỗi lần nhập giá trị cho node, nhấn vào Button “Nhập” để lưu giá trị của node đó. Sau khi hoàn thành việc nhập giá trị cho tất cả các node. Nhấn Button “Đóng” để đóng giao diện nhập liệu.

**A picture containing diagram

Description automatically generated**

Hình 2.1.4. Giao diện sau khi nhập giá trị

Sau khi nhập giá trị, giá trị bên trong các node đã thay đổi như giá trị được nhập từ giao diện nhập liệu. GroupBox “Thuật toán” và “Điều khiển” cũng đã có thể sử dụng.

GroupBox “Thuật toán” gồm có RadioButton “Insertion Sort” dùng để chọn thuật toán sắp xếp chèn.

GroupBox “Điều khiển” chứa 2 RadioButton “Sắp xếp tăng dần” và “Sắp xếp giảm dần” dùng để lựa chọn kiểu sắp xếp. Ngoài ra còn có Button “Sắp xếp” dùng để bắt đầu quá trình sắp xếp.

**Graphical user interface, application

Description automatically generated with medium confidence**

Hình 2.1.5. Các node trong quá trình sắp xếp

Khi sắp xếp sẽ xuất hiện các label trỏ tới node cần chèn và vị trí cần chèn của node đó như là “i=2” là node ở vị trí 2 cần được chèn tới vị trí “POS=1”.

Bubble chart

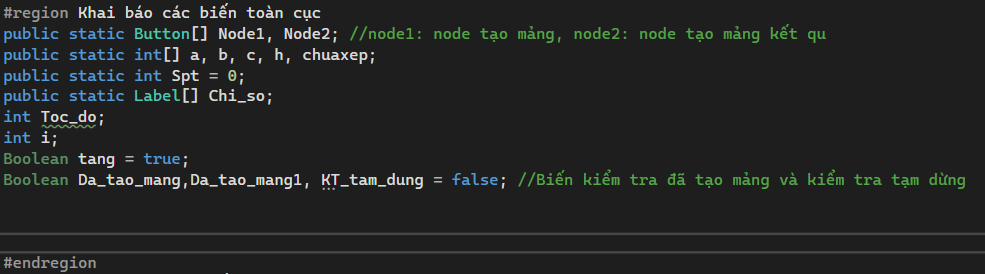
Description automatically generated

Hình 2.1.6. Giao diện sau khi sắp xếp

Sau khi sắp xếp xong, giao diện chính sẽ xuất hiện các label như “Đã sắp xếp xong!”, “Mảng Trước Khi Sắp Xếp”, “Mảng Sau Khi Sắp Xếp”. Cùng với đó là 2 mảng trước khi sắp xếp và sau khi sắp xếp. Bây giờ các GroupBox

1. **Phân tích thuật toán**

Khai báo các biến toàn cục:



Hình 2.2.1. Khai báo biến toàn cục

Khai báo các biến định dạng node:

Text

Description automatically generated

Hình 2.2.2. Khai báo biến định dạng node

1. **Chức năng nút tạo mảng**

**Text

Description automatically generated**

Hình 2.2.3. Chức năng nút tạo mảng

Nếu như trước đó có mảng được sắp xếp, ta ẩn đi mảng ban đầu và mảng được sắp xếp của lần sắp xếp trước đó. Ẩn luôn các label như “GVHD”, “SVTH”, “tb1”, “tb2”, lấy số phần tử của mảng từ TextBox “Sophantu”.

Ta gán biến “Spt” bằng với giá trị bên trong TextBox “Sophantu”. Nếu như giá trị không đúng thì sẽ hiển thị thông báo “Số phần tử nhập vào không hợp lệ” và trả về giá trị của TextBox “Sophantu” là 8.

1. **Tạo mảng:**

****

Hình 2.2.4. Khai báo hàm tạo mảng

Tạo một phương thức Tao\_mang để tạo các mảng cần nhập vào.

Text

Description automatically generated

Hình 2.2.5. Thông báo tạo sai mảng

Text

Description automatically generated Nếu mảng có số phần tử bé hơn 2 và lớn hơn 15, xuất hiện MessageBox “2 <= Số phần tử <= 15”. Xác nhận chưa tạo mảng.

Text

Description automatically generated

Hình 2.2.6. Thiết lập thuộc tính cho node

- Ta thiết lập các thuộc tính cho node ứng với số phần tử trong mảng, nếu mảng có số lượng node từ 2 đến 10, ta đặt kích thước node là 65 pixel, cỡ chữ là 23 pixel, khoảng cách giữa 2 node là 30 pixel, canh lề thì sẽ được tính bằng công thức: x = 1024 – kích thước node \* số phần tử - khoảng cách các node \* (số phần tử - 1) / 2, y = kc

Nếu mảng có số lượng node từ 11 đến 15, ta đặt kích thước node là 50 pixel, cỡ chữ là 18 pixel, khoảng cách các node là 18 pixel, canh lề node được tính tương tự như trên.

Graphical user interface, text

Description automatically generated

Hình 2.2.7. Tạo các mảng dữ liệu

- Ta tạo các mảng dữ liệu bởi các biến Chi\_so, Node1

Text

Description automatically generated

Hình 2.2.8. Tạo node

- Ta tạo node:

Sử dụng vòng lặp for, cho i chạy từ 0 đến số phần từ -1, i tăng dần 1 đơn vị. Ta tạo 1 node như 1 button.

Giá trị bên trong node là biến a ở vị trí i, được căn giữa ở bên trong node. Độ cao và rộng của node được gán bằng biến “kich\_thuoc”, địa điểm của node sẽ được đặt theo công thức: (Canh\_le + (Kich\_thuoc + Khoang\_cach) \* i).

Phông chữ của giá trị sẽ là kiểu chữ đậm, arial và kích thước được gán bằng biến “Co\_chu”.

A screenshot of a computer

Description automatically generated

Hình 2.2.9. Tạo chỉ số

- Chỉ số của node sẽ được tạo như một label:

Các label sẽ được căn giữa với độ cao và rộng bằng biến “Kich\_thuoc”, địa điểm của label sẽ được tính theo công thức: x = (Canh\_le + (Kich\_thuoc + Khoang\_cach) \* i), y = (150 + 3 \* Kich\_thuoc).

Nếu như số phần tử bé hơn hoặc bằng 10: a đặt kiểu chữ Arial, kích thước = “Co\_chu – 10”.

Nếu như số phần tử lớn hơn10: a đặt kiểu chữ Arial, kích thước = “Co\_chu”.

Text

Description automatically generated

Sau khi đã tạo mảng, ta xác nhận đã tạo mảng và cho phép các GroupBox “Nhập” và Button “Nhập” được hoạt động.

1. **Tạo mảng kết quả**

****

Hình 2.2.10. Khai báo hàm tạo mảng kết quả

**-** Tạo một phương thức để tạo mảng kết quả

- Ta tạo thuộc tính cho node trong mảng kết quả giống như lúc ta tạo thuộc tính cho node trong mảng cần sắp xếp.

Text

Description automatically generated

Hình 2.2.11. Tạo node mảng kết quả

- Ta tạo node:

Sử dụng vòng lặp for, cho i chạy từ 0 đến số phần từ -1, i tăng dần 1 đơn vị. Ta tạo 1 node như 1 button.

Độ cao và rộng của button được gán bằng giá trị biến “Kich\_thuoc”.

Địa chỉ của button sẽ được tính theo công thức: x = (Canh\_le + (Kich\_thuoc + Khoang\_cach) \* i, y = 150.

Kiểu chữ của giá trị bên trong node là Arial, chữ đậm, kích thước bằng với giá trị biến “Co\_chu”.

Text

Description automatically generated

Hình 2.2.12. Tạo chỉ số mảng kết quả

- Tạo chỉ số:

Ta tạo các chỉ số như các label, mỗi label đều có chiều cao và chiều rộng bằng với giá trị biến “Kich\_thuoc”.

Địa chỉ của label được tính theo công thức: x = (Canh\_le + (Kich\_thuoc + Khoang\_cach) \* i, y = 400 + 3 \* Kich\_thuoc.

Nếu số phần tử bé hơn hoặc bằng 10, kiểu chữ của label sẽ là Arial, kích thước = Co\_chu – 10.

Nếu số phần tử bé hơn hoặc bằng 10, kiểu chữ của label sẽ là Arial, kích thước = Co\_chu.

A screenshot of a computer

Description automatically generated with medium confidence

Ta xác nhận đã tạo mảng, và cho phép các button “Sắp xếp” và “Nhập” được phép hoạt động.

* + 1. **Xóa mảng**

**Text

Description automatically generated**

Hình 2.2.13. Xóa mảng

Các button “Nhập” và “Sắp xếp” sẽ bị dừng lại. Nếu như mảng ban đầu đã được tạo. Ta chạy vòng lặp for từ 0 đến Spt – 1. Dùng lệnh remove để xóa dần các Node. Ta xác nhận trở thành chưa tạo mảng. Và ta làm tương tự với mảng kết quả.

**2.2.5 Các hàm di chuyển node**

1. **Node qua phải**

Text

Description automatically generated

Hình 2.2.14. Node qua phải

Ta khai báo hai biến “t” dùng để trỏ vào Node và “Step” là số bước dịch chuyển.

Khai báo biến “Loop\_Count” chính là quãng đường dịch chuyển với công thức (Kich\_thuoc + Khoang\_cach) \* Step.

Khi “Loop\_Count” > 0. Ta sẽ dịch chuyển t sang phải 1 đơn vị với tốc độ từ hàm “Tre”, “Loop\_count” giảm dần cho đến 0 thì dừng lại.

1. **Node sang trái**

**Text

Description automatically generated**

Hình 2.2.15. Node qua trái

Với hàm sang trái, ta làm tương tự như hàm sang phải nhưng thay vì t sang phải thì sẽ đổi thành t sang trái.

1. **Node đi lên**

Text

Description automatically generated

Hình 2.2.16. Node đi lên

Ta khai báo 2 biến “t” dùng để trỏ vào Node cần di chuyển và “S” là quãng đường đi lên của Node.

Khai báo “loop\_Count” bằng giá trị của “S”. Khi “loop\_Count” có giá trị lớn hơn 0. “t” sẽ đi lên 1 đơn vị và cứ sau 1 lần dịch chuyển 1 node thì sẽ có thời gian dừng theo hàm “Tre” và “loop\_Count” giảm dần về 0 thì dừng lại.

1. **Node đi xuống**

**Text

Description automatically generated**

Hình 2.2.17. Node đi xuống

Với Node đi xuống, ta làm tương tự với Node đi lên nhưng thay vì “t” đi lên thì ở đây “t” sẽ đi xuống.

**2.2.6 Các hàm khác**

1. **Hàm ngủ**

**Text

Description automatically generated**

Hình 2.2.18. Hàm ngủ

Khai báo biến “millisecond” tức là thời gian mà node cần dịch chuyển. Trong thời gian đó, các node sẽ dừng lại trong vòng 5 mili giây.

1. **Hàm trễ**

**Text

Description automatically generated**

Hình 2.2.19. Hàm trễ

Hàm trễ dùng để gọi lại hàm ngủ.

1. **Hàm hoán đổi vị trí 2 Node**

**Text

Description automatically generated**

Hình 2.2.20. Hàm hoán vị node

Ta khai báo 2 biến “t1” và “t2” là 2 vị trí của một Node.

Ta cho Node1 ở vị trí **“**t1” như một button tạm. Nếu như Node1 được chuyển từ vị trí “t1” sang “t2” thì Node1[t2] trở thành button tạm.

1. **Text

   Description automatically generatedText

   Description automatically generatedHàm hoàn thành**

Hình 2.2.21. Hàm hoàn thành

Khi thực hiện hàm “hoàn thành sắp xếp”, ta sẽ ẩn các mũi tên chỉ vào Node và vị trí cần đến, xuất hiện label với dòng chữ “Đã sắp xếp xong” màu tím. Các label như “Mảng trước khi sắp xếp” và “Mảng sau khi sắp xếp” cũng sẽ xuất hiện. Button “Tao\_mang” cũng sẽ được cho phép sử dụng. Ta xóa đi mảng được xếp và tạo mảng kết quả, tạo mảng trước khi sắp xếp. Button “Nhap” sẽ bị vô hiệu hóa cùng với đó là các RadioButton như “insertionsort”, “giam”, “tang” cũng sẽ bị vô hiệu hóa.

**2.2.7 Nhập dữ liệu**

**A screenshot of a computer

Description automatically generated with medium confidence**

Hình 2.2.22. Chức năng nút nhập dữ liệu

- Ta tạo chức năng cho button “Nhap”:

Khi ta nhấn vào button “Nhap” thì button “Sapxep” sẽ được cho phép chọn cũng như là các RadioButton như “insertionsort”, “giam”, “tang” sẽ được phép chọn.

Form nhập liệu cũng được mở ra.

**2.2.9 Hàm sắp xếp**

1. **Sắp xếp tăng dần**

**Text

Description automatically generated**

Ta khai báo các biến như “i”, “pos”, “x”, “Chi\_so\_tam”, “Dem\_node” và một button “Node\_tam”.

Text

Description automatically generated

Ta cho “i” chạy từ 1 đến Spt – 1, i tăng dần 1 đơn vị. Ta gọi hàm “Tre” để có khoảng nghỉ giữa các lần di chuyển.

Ta thiết lập node ở vị trí 0 là node đã được xếp, thiết lập màu cho node là màu đen.

Ta gán “Dem\_node” bằng 0, “x” bằng giá trị của node a ở vị trí “i”, “Node\_tam” sẽ có giá trị của “Node1” ở vị trí “i”. “Chi\_so\_tam” cũng sẽ bằng “i”, và “pos” có giá trị “i – 1”. Những bước này dùng để đếm số bước dịch chuyển của 1 node.

Ta thiết lập các mũi tên đánh dấu nút cần chèn bằng cách hiện các mũi tên, vị trí của mũi tên sẽ được tính theo công thức: x = ((Canh\_le + (Kich\_thuoc + Khoang\_cach) \* i) + (Kich\_thuoc / 2) – 30; y = Node1[i].Location.Y – Kich\_thuoc -70. Bên trong mũi tên sẽ chứa dòng chữ “i = [giá trị của i]”

Sau đó ta di chuyển node được chọn lên trên bằng cách gọi hàm “Node\_di\_len” với node được chọn gán với “Node\_tam” và quãng đường là (Kich\_thuoc) + 5.

Text

Description automatically generated

Ta tìm vị trí cần chèn cho node. Khi giá trị của “pos” lớn hơn hoặc bằng 0 và giá trị node tại “pos” lớn hơn giá trị node di chuyển lên. Ta thiết lập mũi tên chỉ vào vị trí đó với công thức chọn vị trí là: X = ((Canh\_le + (Kich\_thuoc + Khoang\_cach) \* pos) + (Kich\_thuoc / 2) – 30, Y = Node1[pos].Location.Y + Kich\_thuoc + 5. Bên trong mũi tên có dòng chữ “POS = [vị trí node]”.

Sau đó ta cho node tại vị trí “pos + 1” sẽ có giá trị của node tại vị trí “pos” và đưa biến “Dem\_node” tăng 1 đơn vị. Sau đó ta gọi hàm “Node\_qua\_phai” với đối tượng là Node1[pos] với số bước là 1.

Text

Description automatically generated

Sau đó ta gọi hàm hoán vị node ở vị trí “pos + 1” và “pos”. Và “pos” giảm 1 đơn vị. Gán giá trị a[pos + 1] bằng với “x”. Sau đó ta gọi hàm “Node\_qua\_trai” với đối tượng là “Node\_tam” và số bước là “Dem\_node”. Ta tiếp tục gọi hàm “Node\_di\_xuong” với đối tượng là “Node\_tam” và quãng đường đi là: (Kich\_thuoc + 5).

Ta tắt đi mũi tên. Rồi gán Node ở vị trí “pos + 1” sẽ là “Node\_tam” tiếp theo. Sau đó định dạng cho node này là node đã xếp và có màu đen. Sau cùng ta gọi hàm “Hoan\_thanh” để dừng việc xếp.

1. **Sắp xếp giảm dần**

**Text

Description automatically generated**

Đối với hàm sắp xếp giảm dần, ta cũng khai báo các biến giống như sắp xếp tăng dần. Ta cũng thiết lập node đầu tiên là node đã được sắp xếp. Cách tính số bước dịch chuyển và thiết lập mũi tên đánh dấu cần chèn cũng tương tự.

A screenshot of a computer

Description automatically generated with medium confidence

Ta cũng cho node được chọn sẽ đi lên. Ta tìm vị trí cần chèn node thì thay vì ở sắp xếp tăng dần giá trị của a[pos] lớn hơn “x” thì ở đây giá trị sẽ là bé hơn “x”. Ta thiết lập mũi tên chỉ vào vị trí cần chèn node cũng tương tự. Sau đó ta dịch chuyển node ở vị trí “pos” sang phải.

Text

Description automatically generated

Sau đó ta đưa node được đưa lên trên sang trái và đi xuống chỗ được chèn. Thiết lập Node1[pos + 1] là “Node\_tam” và định dạng node này là đã được xếp và có màu đen. Sau cùng ta gọi hàm “Hoan\_thanh” để dừng việc sắp xếp.

**2.2.9 Chức năng nút sắp xếp**

**Text

Description automatically generated**

Hình 2.2.23. Chức năng nút sắp xếp

Sau khi button “sapxep” được nhấn, ta sẽ gán số lượng node chưa xếp bằng với “Spt” và các node đó đều chưa được xếp. Lúc này các button “Sapxep”, “Taomang”, “Nhap” sẽ không được cho phép chọn.

Nếu như RadioButton “insertionsort” và “tang” được chọn thì sẽ gọi hàm “InsertionSort\_tang” và ngược lại thì sẽ gọi hàm “InsertionSort\_giam”.

# **CHƯƠNG 3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỀN**

## 3.1 Kết luận

Thuật toán sắp xếp chèn là một trong các thuật toán nổi tiếng đối với những lập trình viên trên khắp thế giới. Việc mô phỏng lại quá trình sắp xếp chèn giúp cho chúng ta dễ hình dung hơn phương thức hoạt động của thuật toán. Sau một thời gian tìm hiểu, nghiên cứu, lập trình thì đề tài cũng đã được mô phỏng trên giao diện WinForm của ngôn ngữ C#. Giao diện trình bày được đầy đủ các chức năng, tiến trình của việc sắp xếp chèn. Các công đoạn được trình bày cụ thể, dễ hiểu, dễ nắm bắt, thông tin được đưa ra rõ ràng, dễ nhìn. Khi quy trình sắp xếp chèn được kích hoạt, ta có thể sắp xếp được các mảng theo một trật tự nhất định, có thể là tăng dần hoặc giảm dần.

## 3.2 Hướng phát triển

Hướng phát triển của đề tài sau này là mô phỏng thêm được các thuật toán sắp xếp khác mang tính phức tạp cao hơn. Cũng như sẽ bổ sung thêm các tính năng như làm chậm, tạm dừng nhằm hiểu rõ hơn về tiến trình sắp xếp. Thay đổi giao diện sao cho phù hợp, dễ nhìn hơn.

# **TÀI LIỆU THAM KHẢO**

[1] <https://vi.wikipedia.org/wiki/S%E1%BA%AFp_x%E1%BA%BFp_ch%C3%A8n>

[2] <https://www.geeksforgeeks.org/insertion-sort/>

[3] <https://sharecode.vn/>

[4] <https://viblo.asia/p/cac-thuat-toan-sap-xep-co-ban-Eb85ooNO52G>

[5] <https://tailieumienphi.vn/doc/de-tai-cong-nghe-thong-tin-cac-thuat-toan-sap-xep-co-ban-irsauq.html>