**MỤC LỤC**

[I. PHẦN MỞ ĐẦU](#_Toc25952821) 2

[1. Lí do và mục đích chọn đề tài](#_Toc25952822) 2

[2. Mục tiêu cho đề tài](#_Toc25952823) 2

[3. Giới thiệu đề tài Game Sudoku](#_Toc25952824) 2

[II. CƠ SỞ LÝ THUYẾT](#_Toc25952825) 3

[1. Ngăn xếp (stack) là gì?](#_Toc25952826) 3

[2. Khởi tạo stack](#_Toc25952827) 3

[2.1. Định nghĩa kiểu dữ liệu stack](#_Toc25952828) 3

[2.2. Thêm 1 phần tử vào ngăn xếp (push)](#_Toc25952829) 3

[2.3. Lấy 1 phần tử ra khỏi danh sách (pop)](#_Toc25952830) 3

[3. Một số ứng dụng của ngăn xếp](#_Toc25952835) 4

[3.1. Đảo ngược xâu ký tự](#_Toc25952836) 4

[3.2. Tính giá trị biểu thức dạng hậu tố](#_Toc25952837) 4

[III. PHẦN NỘI DUNG](#_Toc25952838) 6

[1. Mô tả ứng dụng](#_Toc25952839) 6

[1.1. Chức năng phần mềm.](#_Toc25952840) 6

[1.2. Các chức năng thêm](#_Toc25952845) 6

[2. Quá trình thực viết chương trình](#_Toc25952847) 7

[2.1. Giao diện của các form trong chương trình](#_Toc25952848) 7

[2.2. Các class trong chương trình](#_Toc25952849) 8

[2.3. Code áp dụng ngăn xếp trong chương trình](#_Toc25952850) 8

[IV. KẾT LUẬN](#_Toc25952851) 10

[1. Bảng phân công kế hoạch](#_Toc25952852) 10

[2. Đánh giá ưu và nhược điểm của quá trình làm](#_Toc25952853) 11

[2.1. Tự đánh giá](#_Toc25952854) 11

[2.2. Ưu điểm](#_Toc25952856) 11

[2.3. Nhược điểm](#_Toc25952857) 11

[2.4. Hướng phát triển](#_Toc25952857) 12

[V. TÀI LIỆU THAM KHẢO](#_Toc25952858) 13

**DANH MỤC HÌNH**

[Hình 1: Giao diện chính của chương trình](#_Toc25254756) 7

[Hình 2: Giao diện tạo bảng Sudoku mới](#_Toc25254757) 7

[Hình 3: Cấu trúc thư mục chương trình](#_Toc25254757) 8

**DANH MỤC BẢNG**

[Bảng 1: Code áp dụng ngăn xếp trong chương trình](#_Toc25254851) 9

[Bảng 2: Bảng phân công kế hoạch](#_Toc25254851) 10

# LỜI CẢM ƠN

Trong thời gian làm đồ án môn cấu trúc dữ liệu, nhóm đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy và bạn bè. Nhóm xin gửi lời cảm ơn chân thành đến Th.S Trần Công Tú, giảng viên phụ trách môn Cấu trúc dữ liệu và giải thuật người đã tận tình hướng dẫn, chỉ bảo nhóm trong suốt quá trình làm đồ án.

Để hoàn thành đồ án “GAME SUDOKU SỬ DỤNG NGĂN XẾP”, nhóm đã khảo sát, thu nhập dữ liệu và tổng hợp thông tin từ nhiều hướng khác nhau. Nhưng quá trình xây dựng chương trình, nghiên cứu và cố gắng hoàn thành, nhưng vì trình độ còn hạn chế nên có nhiều khó khăn làm chậm tiến độ. Mặt khác, đề tài tuy đã có sự giúp đỡ của GVHD nhưng vẫn không thể tránh khỏi những sai sót.

Do đó, nhóm mong sẽ nhận được sự góp ý chân thành của thầy và các bạn đọc. Những ý kiến đóng góp về đề tài sẽ được nhóm tiếp nhận để có thêm kinh nghiệm khi làm các đề tài nghiên cứu khác. Một lần nữa, nhóm xin cảm ơn thầy Trần Công Tú và các bạn!

1. **PHẦN MỞ ĐẦU**
2. **Lí do và mục đích chọn đề tài**

Dựa vào những kiến thức đã học từ những môn nhập môn lập trình và kỹ thuật lập trình , nhóm muốn tạo ra một game Sudoku đơn giản kết hợp với window form,..

Mục đích chính khi nhóm chọn đề tài này vì : Muốn tạo một ra ứng dụng giải trí áp dụng kiến thức đã học đồng thời sau đó có thể cải thiện khả năng tư duy logic, tính toán.

Từ những đặc điểm trên nên nhóm quyết định xây dựng ứng dụng game Sudoku sử dụng ngăn xếp.

1. **Mục tiêu cho đề tài**

Áp dụng được ngăn xếp vào phần mềm để có thể thực hiện chức năng undo và redo. Bên cạnh đó phần mềm còn có những chức năng như: Hiển thị đáp án, kiểm tra đúng sai.

1. **Giới thiệu đề tài Game Sudoku**

Game Sudoku từ lâu đã được mọi người biết đến và yêu thích. Nhưng trong thời đại công nghệ 4.0 hiện nay, thị trường đã có nhiều loại game 3D với đồ họa đẹp và bắt mắt nhưng game Sudoku vẫn còn được ưa thích bởi giao diện và lối chơi đơn giản cũng như lợi ích mà nó mang lại là tính giải trí cùng với đó là tăng khả năng tư duy logic cho người chơi. Và game Sudoku cũng được nhóm xây dựng trên công nghệ thông qua ngôn ngữ lập trình winform C#.

1. **CƠ SỞ LÝ THUYẾT**

## **1. Ngăn xếp (stack) là gì?**

Ngăn xếp là 1 dạng đặc biệt của danh sách liên kết mà việc bổ sung hay loại bỏ 1 phần tử đều thực hiện ở 1 đầu của danh sách gọi là đỉnh.

Ngăn xếp có 2 thao tác cơ bản: thêm phần tử vào được gọi là push và loại bỏ phần tử được gọi là pop.

Việc loại bỏ phần tử sẽ tiến hành loại bỏ phần tử mới nhất được đưa vào danh sách, chính vì tính chất này mà ngăn xếp còn được gọi là kiểu dữ liệu LIFO (Last in first out – Vào sau ra trước).

## **2. Khởi tạo stack**

### **2.1. Định nghĩa kiểu dữ liệu stack**

Vì stack là 1 dạng đặc biệt của danh sách liên kết nên ta có thể dùng kiểu dữ liệu Node đã trình bày ở bài danh sách liên kết để biểu diễn kiểu dữ liệu của stack.

Định nghĩa 1 class Stack với kiểu dữ liệu generic T và 1 node trên cùng gọi là top.

### **2.2. Thêm 1 phần tử vào ngăn xếp (push)**

Kiểm tra xem ngăn xếp có nil không, nếu nil thì gán đỉnh ngăn xếp vào phần tử muốn thêm vào.

Nếu đỉnh ngăn xếp không nil, tạo 1 nút mới, gán phần tử muốn thêm vào nút đó, gán đỉnh ngăn xếp vào nút vừa tạo.

### **2.3. Lấy 1 phần tử ra khỏi danh sách (pop)**

Kiểm tra danh sách đỉnh có rỗng không, nếu rỗng thì kết thúc.

Nếu đỉnh danh sách không rỗng, kiểm tra nút đỉnh trỏ đến có rổng không nếu rỗng thì gán đỉnh danh sách bằng nil (trường hợp danh sách chỉ có 1 phần sau khi lấy 1 phần tử ra thì danh sách trở thành rỗng), còn không gán đỉnh của danh sách vào nút tiếp theo.

## **3. Một số ứng dụng của ngăn xếp**

### **3.1 Đảo ngược xâu ký tự**

Bài toán đảo ngược xâu ký tự yêu cầu hiển thị các ký tự của 1 xâu ký tự theo chiều ngược lại.

Ký tự cuối cùng của xâu sẽ được hiển thị trước, tiếp theo là ký tự sát ký tự cuối… và ký tự đầu tiên sẽ được hiển thị đầu tiên.

Để giải quyết bài toán, ta chỉ cần duyệt từ đầu đến cuối xâu, lần lượt cho các ký tự vào ngăn xếp.

Khi đó, các ký tự đầu tiên sẽ vào trước, tiếp theo đến ký tự thứ 2 … ký tự cuối cùng vào sau cùng. Sau khi đã cho toàn bộ ký tự của xâu vào ngăn xếp, lần lượt lấy các phần tử ra khỏi ngăn xếp và hiển thị lên màn hình. [1]

### **3.2 Tính giá trị biểu thức dạng hậu tố**

Một biểu thức toán học thông thường bao gồm các toán tử: + - \* / , các toán hạng và dấu ngoặc cho biết thứ tự tính toán.

Chẳng hạn 1 biểu thức toán học: (3 \* (((5 - 2) \* (7 + 1)) - 6)). Như chúng ta đã thấy trong biểu thức trên, các toán tử bao giờ cũng nằm giữa 2 toán hạng. Do vậy, cách viết trên được gọi là cách viết trung tố (infix). Để tính toán giá trị biểu thức trên, ta phải tính toán giá trị các phép toán trong ngoặc trước. Đôi khi ta cần lưu các kết quả tính toán này như 1 kết quả trung gian, sau đó sử dụng chúng như toán hạng tiếp theo. Ví dụ ở biểu thức trên, đầu tiên ta tính 5 – 2 = 3, lưu kết quả này lại. Trong các biểu thức dạng này, vị trí dấu ngoặc rất quan trọng. Nếu vị trí các dấu ngoặc thay đổi thì giá trị biểu thức cũng thay đổi theo. Đối với con người, cách trình bày biểu thức toán học theo dạng này có vẻ như là hợp lý nhất, nhưng đối với máy tính, việc tính toán nhũng biểu thức như vậy là tương đối phức tạp. Để dễ dạng cho máy tính trong việc tính toán các biểu thức, người ta đưa ra 1 cách trình bày khác cho biểu thức toán học, đó là dạng hậu tố (postfix). [1]

Theo cách trình bày này, toán tử không nằm giữa 2 toán hạng mà nằm ở ngay phía sau 2 toán hạng. Chẳng hạn ở biểu thức trên ta có thể viết dưới dạng hậu tố như sau: 3 5 2 – 7 1 + \* 6 - \*. Toán tử - nằm ngay sau 2 toán hạng 5 và 2 nên lấy 5 – 2 = 3, lưu kết quả 3. Toán tử + nằm ngay sau 2 toán hạng 7 và 1 nên lấy 7 + 1 = 8, lưu kết quả 8. Toán tử \* nằm ngay sau 2 kết quả vừa lưu nên lấy 3 \* 8 = 24, lưu kết quả 24. Toán tử - nằm ngay sau kết quả vừa lưu và 6 nên lấy 24 – 6 = 18. Toán tử \* nằm ngay sau kết quả vừa lưu và 3 nên lấy 3 \* 18 = 54 là kết quả cuối cùng của biểu thức.

1. **PHẦN NỘI DUNG**
2. **Mô tả ứng dụng**
   1. **Chức năng phần mềm**.

- Phần mềm có chức năng cơ bản là giải ô đố Sudoku.

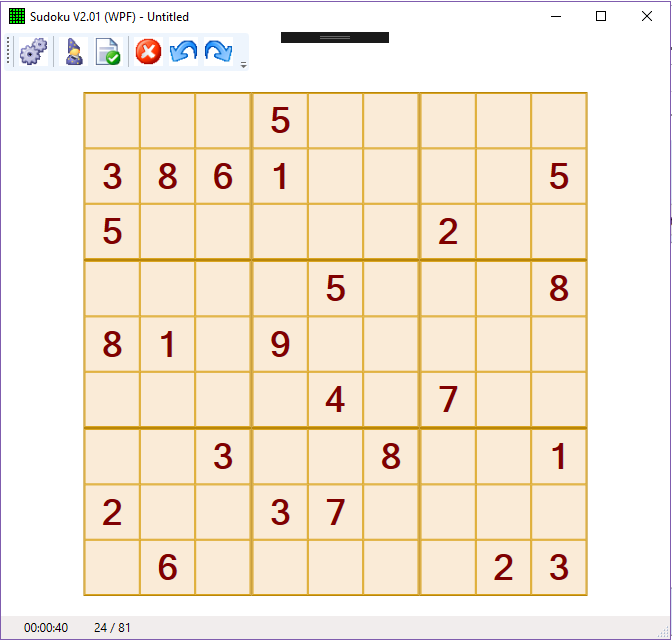
- Show đáp án của tất cả của các ô.

- Undo, redo.

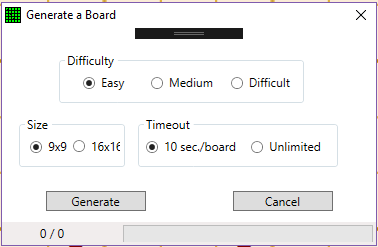
- Check đúng sai của các ô đã điền.

* 1. **Các chức năng thêm**

Ngoài các chức năng trên thì phần mềm còn có chức năng tùy chọn: loại bảng số 9x9 hoặc bảng chữ 16x16, độ khó, timeout …

1. **Quá trình thực viết chương trình**
   1.  **Giao diện của các form trong chương trình:**

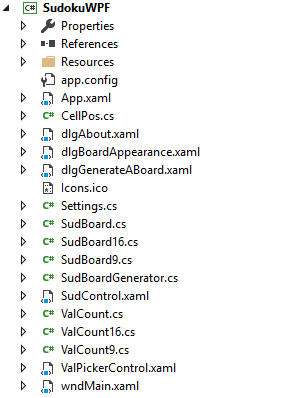
**Hình 1:** Giao diện chính của chương trình



**Hình 2:** Giao diện tạo bảng Sudoku mới

* 1. **Các class trong chương trình:**

Cấu trúc thư mục trong chương trình:



**Hình 3:** Cấu trúc thư mục chương trình

* 1. **Code áp dụng ngăn xếp trong chương trình:**

|  |
| --- |
| /// Undo a move  ///  ///ptCellPos : Cell position  /// bRedoable : True if the undo move can be redo  ///  public bool UndoMove(out CellPos ptCellPos, bool bRedoable) {  bool bRetVal;  int iPos;    bRetVal = UndoMove(out iPos, bRedoable);  ptCellPos = new CellPos(iPos % m\_iLineSize, iPos / m\_iLineSize);  if (!bRedoable) {  m\_stackRedoMoves.Clear();  }  return(bRetVal);  }  /// Redo a move  ///  /// ptCellPos : Cell position  ///  public bool RedoMove(out CellPos ptCellPos) {  bool bRetVal;  int iPos;  int iOldVal;  int iVal;  Move move;    ptCellPos = new CellPos();  if (m\_stackRedoMoves.Count == 0) {  bRetVal = false;  } else {  move = m\_stackRedoMoves.Pop();  iPos = move.m\_nPos;  iVal = move.m\_nOldVal;  ptCellPos.X = iPos % m\_iLineSize;  ptCellPos.Y = iPos / m\_iLineSize;  iOldVal = m\_arrBoardVal[iPos];  bRetVal = PlayAt(iPos,  iVal,  true, // User move  false, // Don't record the move  false); // No validation  if (bRetVal) {  m\_stackUndoMoves.Push(new Move(iPos, iOldVal));  }  }  return(bRetVal);  } |

**Bảng 1:** Code áp dụng ngăn xếp trong chương trình

1. **KẾT LUẬN**
2. **Bảng phân công kế hoạch**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Danh sách công việc | Nguyễn Huy Cường | Phạm Thanh Trung | Ngày bắt đầu (dự kiến) | Ngày kết thúc (dự kiến) | Ngày bắt đầu (thực tế) | Ngày kết thúc (thực tế) |
| Thiết kế giao diện |  | X | 11/09/2019 | 18/09/2019 | 11/09/2019 | 18/09/2019 |
| Triển khai | X |  | 19/09/2019 | 26/09/2019 | 19/09/2019 | 26/09/2019 |
| Tìm hiểu ngăn xếp | X | X | 27/09/2019 | 04/10/2019 | 27/09/2019 | 04/10/2019 |
| Xây dựng game Sudoku | X | X | 05/10/2019 | 12/10/2019 | 05/10/2019 | 15/10/2019 |
| Tạo bảng Sudoku | X | X | 15/10/2019 | 20/10/2019 | 12/10/2019 | 22/10/2019 |
| Xây dựng chúc năng hiển thị kết quả | X |  | 22/10/2019 | 27/10/2019 | 22/10/2019 | 31/10/2019 |
| Xây dựng chức năng kiểm tra kết quả |  | X | 01/11/2019 | 07/11/2019 | 01/11/2019 | 09/11/2019 |
| Xây dựng chức năng Undo và Redo | X | X | 10/11/2019 | 20/11/2019 | 10/11/2019 | 22/11/2019 |
| Hoàn chỉnh code | X | X | 23/11/2019 | 25/11/2019 | 23/11/2019 | 25/11/2019 |
| Hoàn chỉnh báo cáo | X | X | 26/11/2019 | 28/11/2019 | 26/11/2019 | 29/11/2019 |

**Bảng 2:** Bảng phân công kế hoạch

1. **Đánh giá ưu điểm và nhược điểm của quá trình làm**
   1. **Tự đánh giá:**

Phần mềm đáp ứng khá đủ yêu cầu ban đầu đưa ra, đã tạo được những chức năng cần thiết của một game Sudoku cơ bản. Tuy nhiên giao diện vẫn còn khá thô sơ, chưa hoàn thiện, chưa tạo được chức năng gợi ý cũng như những chức năng nâng cao khác.

* 1. **Ưu điểm:**

- Ứng dụng cho phép người chơi tạo bảng số 9x9 hoặc bảng chữ 16x16 tùy theo nhu cầu.

- Ta có thể undo (đi lùi) hoặc redo (đi tới) nếu muốn thay đổi giá trị ô số đã nhập.

- Ứng dụng cũng cho phép ta show ra đáp án của tất cả ô số cũng như check xem các ô số đã điền có đúng hay không.

* 1. **Nhược điểm**

- Ở chế độ chơi bảng chữ 16x16, phần chọn chữ còn nhỏ, khó click chọn.

- Phần giao diện thiết kế chưa được đẹp, chưa hoàn thiện.

* 1. **Hướng phát triển:**

- Thêm tính năng gợi ý.

- Thay đổi cách chơi: thay vì click chọn số thì cho người chơi nhập số (hoặc chữ) từ bàn phím.

- Thêm phần điểm cao tính theo thời gian người chơi hoàn thành một game.

1. **TÀI LIỆU THAM KHẢO**

[1] Cấu trúc dữ liệu và giải thuật: Ngăn xếp (Dec 26th, 2016) <https://viblo.asia/p/cau-truc-du-lieu-va-giai-thuat-ngan-xep-stack-DbmemoJqvAg>