

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



BÁO CÁO ĐỒ ÁN MÃ NGUỒN MỞ

ĐỀ TÀI

ỨNG DỤNG EASYTYPE
(Text Editor)

Môn học	: Đồ án mã nguồn mở
Giảng viên	: Đỗ Thị Thanh Tuyền
Nhóm thực hiện	: Nguyễn Ngọc Nghĩa - 16521585
	Dương Thanh Tín - 16521241
	Hồ Nguyễn Nhật Tiến - 16521218

TP. Hồ Chí Minh, tháng 12 năm 2019

LỜI NÓI ĐẦU

EasyType được thiết kế dựa trên ứng dụng mã nguồn mở Notepad++, là một công cụ soạn thảo code nổi tiếng được nhiều người biết đến và sử dụng không chỉ bởi dung lượng nhẹ, dễ cài đặt, sử dụng mà còn nhớ đến những chức năng đa dạng mà nó cung cấp như Syntax Highlight, Code Folding, Auto Add Matching Brace, Document Map,...

LỜI CẢM ƠN

Nhóm em xin trân trọng gửi lời cảm ơn tới cô Đỗ Thị Thanh Tuyền đã tạo kiện về thời gian để chúng em có thể làm được đồ án môn học lần này.

Chỉ trong vòng 6 tuần, chúng em đã tìm tòi và học hỏi được những kiến thức quan trọng, thiết yếu để có thể làm được một chương trình hoàn chỉnh.

Cũng xin cảm ơn thầy cô trong khoa Công nghệ phần mềm khác đã nhiệt tình hỗ trợ, giải đáp những thắc mắc cho chúng em về bài báo cáo đồ án này.

MỤC LỤC

LỜI NÓI ĐẦU	2
LỜI CẢM ƠN	3
MỤC LỤC	4
I. GIỚI THIỆU	5
1. Lý do chọn đề tài.....	5
2. Mục đích nghiên cứu đề tài.....	5
3. Phạm vi nghiên cứu đề tài.....	5
4. Công nghệ sử dụng	5
II. CÁC CHỨC NĂNG CHÍNH	6
1. Syntax Highlighting	6
2. Line Number	6
3. Document Map.....	6
4. Tìm kiếm và thay thế.....	6
5. Auto Complete Word	6
6. Brace Matching and Auto Indenting	6
7. Multi-Document	6
III. THIẾT KẾ.....	6
1. Thiết kế giao diện.....	6
2. Thiết kế kiến trúc chương trình.....	9
IV. CÀI ĐẶT VÀ THỬ NGHIỆM	18
1. Môi trường và ngôn ngữ cài đặt.....	18
2. Thử nghiệm	18
V. ĐÁNH GIÁ KẾT QUẢ.....	19
1. Kết quả đạt được	19
2. Thuận lợi và khó khăn	19
VI. KHÁC	20
1. Phân công công việc	20
2. Hướng phát triển phần mềm	20
3. Tài liệu tham khảo	20

I. GIỚI THIỆU

1. Lý do chọn đề tài

Nhóm chọn đề tài này nhằm hiểu rõ các thao tác xử lý văn bản và có thể sử dụng lại chính phần mềm của mình để viết phần mềm khác.

Ngoài ra, đề tài Ứng dụng EasyType không quá dễ, giúp cho nhóm có động lực để có thể làm việc nhóm một cách nghiêm túc, đồng thời củng cố kiến thức môn học.

2. Mục đích nghiên cứu đề tài

Tìm hiểu cách xử lý văn bản và các kỹ thuật liên quan đến viết một chương trình code editor đơn giản, có thể đưa vào sử dụng.

3. Phạm vi nghiên cứu đề tài

Giới hạn các chức năng có trong sản phẩm chỉ hoạt động đúng mục đích của chính nó, chưa cần phải thông minh. Các ngôn ngữ hỗ trợ là C, C++, C#, Javascript, VB, SQL.

4. Công nghệ sử dụng

Chương trình sử dụng công nghệ Win Forms App với ngôn ngữ C#.

Mã nguồn được viết trên Visual Studio 2017.

II. CÁC CHỨC NĂNG CHÍNH

1. Syntax Highlighting

Syntax Highlighting là một trong những chức năng quan trọng nhất trong việc coding cho lập trình viên. Cụ thể chức năng này sẽ làm đổi màu những từ hoặc những đoạn text mang ý nghĩa đặc biệt tùy thuộc vào ngôn ngữ mà lập trình viên sử dụng. Điều này giúp lập trình viên dễ dàng hơn trong việc phân biệt cấu trúc của code, dễ tìm và sửa lỗi.

2. Line Number

Đây là chức năng quan trọng trong một chương trình soạn thảo văn bản cho lập trình viên. Giúp cho lập trình viên có thể dễ dàng theo dõi số hàng của chương trình cũng như tìm lỗi, đánh dấu, sửa lỗi.

3. Document Map

Document Map cung cấp cho người dùng một giao diện thu nhỏ của phần soạn thảo văn bản để lập trình viên có thể nắm được bố cục của toàn bộ phần code, giúp dễ dàng phát hiện lỗi trình bày và chỉnh sửa.

4. Tìm kiếm và thay thế

Chức năng không thể thiếu cho lập trình viên hỗ trợ việc tìm kiếm và thay thế hoạt loạt văn bản và từ khóa, tiết kiệm thời gian cho việc lập trình.

5. Auto Complete Word

Hỗ trợ tốc độ lập trình của người dùng bằng cách gợi ý những từ khóa có thể là từ khóa đang gõ và tự hoàn thành nó nếu muốn. Một chức năng không thể thiếu đối với chương trình code editor ngày nay.

6. Brace Matching and Auto Indenting

Đây là một chức năng không thể thiếu trong một chương trình hỗ trợ lập trình. Tự động thêm dấu đóng ngoặc và tự động thụt hàng nếu cần thiết là chức năng giúp lập trình viên tạo ra được những đoạn code dễ đọc, dễ chỉnh sửa mà lại ít tốn thời gian nhất.

7. Multi-Document

Hỗ trợ người dùng có thể mở nhiều Tab chứa các trang văn bản cần thiết để thao tác. Tạo sự thuận tiện và linh hoạt cho việc lập trình.

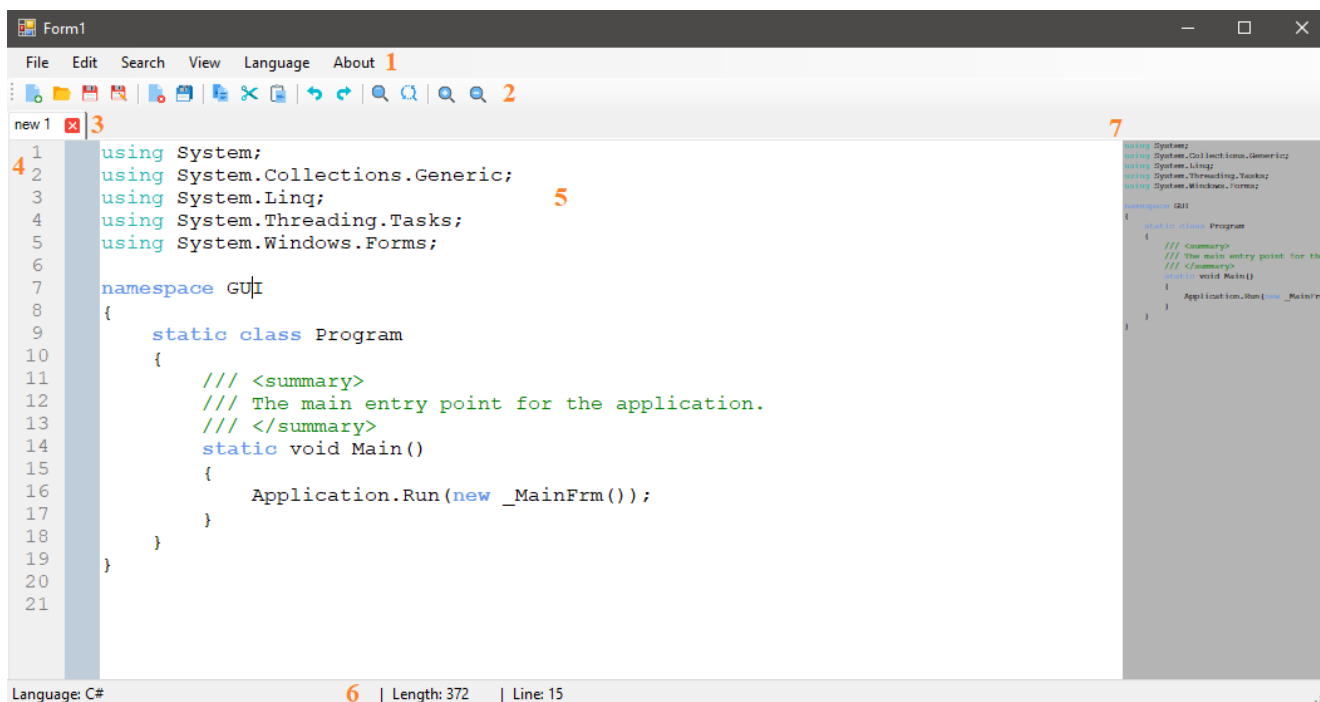
III. THIẾT KẾ

1. Thiết kế giao diện

1.1. Màn hình giao diện chính

Ý tưởng: Màn hình giao diện chính được thiết kế như một phần mềm code editor truyền thống với giao diện tối giản, dễ dàng dùng được ngay mà không cần hướng dẫn.

Hiện thực:

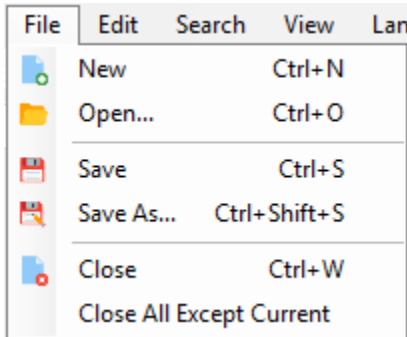


Bảng giải thích các số bên trên:

Số	Tên gọi	Chức năng
1	Thanh công cụ (Menu bar)	Cung cấp các thao tác mà chương trình hỗ trợ
2	Thanh công cụ truy cập nhanh (Tool Bar)	Cung cấp các nút gọi chức năng mà chương trình hỗ trợ một cách nhanh chóng
3	Thanh quản lý trang (Tab Bar)	Quản lý các trang đã trang đã được mở (đóng hoặc đổi vị trí)
4	Thanh đánh số dòng (Line Number)	Hiển thị số thứ tự của các dòng để người dùng dễ theo dõi
5	Phần soạn thảo văn bản (Text Area)	Nhận dữ liệu từ bàn phím để hiển thị văn bản
6	Thanh trạng thái (Status Bar)	Hiển thị một số trạng thái của trang hiện tại (Độ dài, Số dòng, Ngôn ngữ đang sử dụng)
7	Document Map	Dùng để xem sơ lược toàn bộ những gì người dùng đã nhập trong trang

1.2. Thanh công cụ (Menu Bar)

2.1. File



New: Tạo một tab mới chứa trang văn bản.

Open: Mở một tab mới với trang văn bản có sẵn.

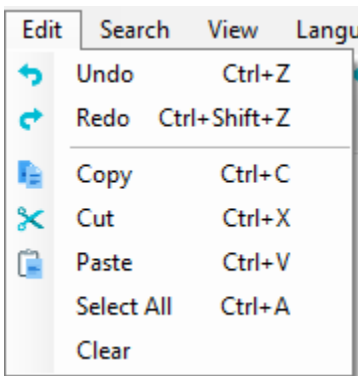
Save: Lưu văn bản ở tab hiện tại xuống máy tính theo đường dẫn tùy ý người dùng (Lưu đề lên nếu văn bản đã được lưu trước đó).

Save As: Lưu văn bản ở tab hiện tại xuống máy tính theo đường dẫn tùy ý người dùng (Không lưu đề lên văn bản đã lưu trước đó).

Close: Tắt tab hiện tại.

Close All Except Current: Tắt toàn bộ tab trừ tab hiện tại.

2.2. Edit



Undo: Thu hồi hành động vừa mới thực hiện

Redo: Thực hiện lại hành động vừa mới thu hồi

Copy: Sao chép đoạn văn bản được chọn

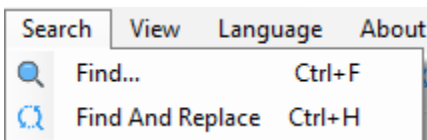
Cut: Cắt đoạn văn bản được chọn

Paste: Dán đoạn văn bản đã Copy hoặc Cut.

Select All: Bôi đen toàn bộ văn bản

Clear: Xóa toàn bộ văn bản

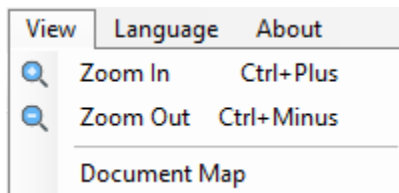
2.3. Search



Find: Tìm kiếm từ theo ý người dùng

Find And Replace: Tìm kiếm và thay thế từ theo ý người dùng.

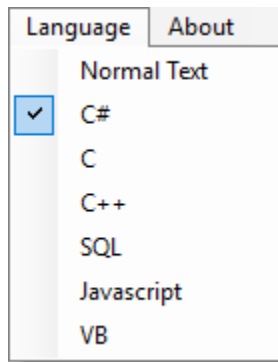
2.4. View



Zoom In: Phóng to trang văn bản

Zoom Out: Thu nhỏ trang văn bản

2.5. Language



Normal Text: Soạn thảo văn bản thông thường.

C#: Soạn thảo văn bản với ngôn ngữ C#.

C: Soạn thảo văn bản với ngôn ngữ C.

C++: Soạn thảo văn bản với ngôn ngữ C++.

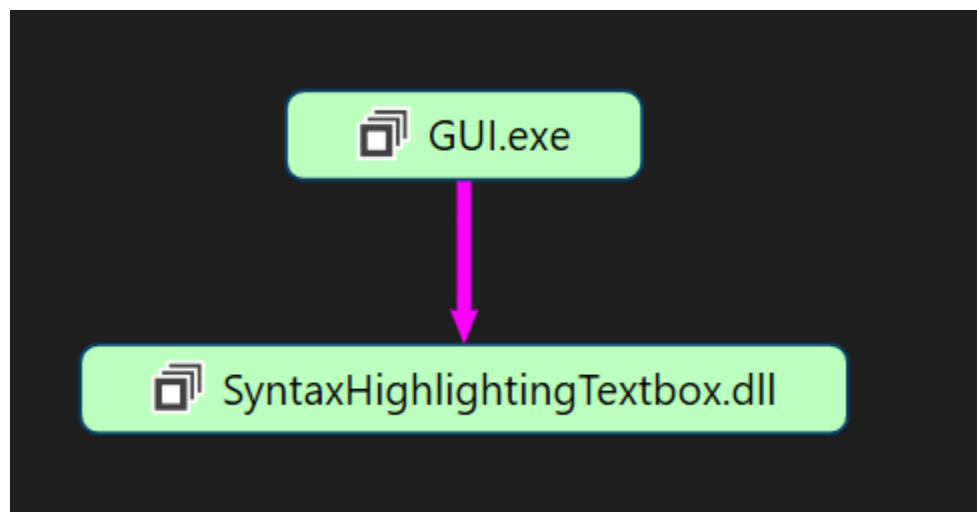
SQL: Soạn thảo văn bản với ngôn ngữ SQL.

Javascript: Soạn thảo văn bản với ngôn ngữ Javascript.

VB: Soạn thảo văn bản với ngôn ngữ VB.

2. Thiết kế kiến trúc chương trình

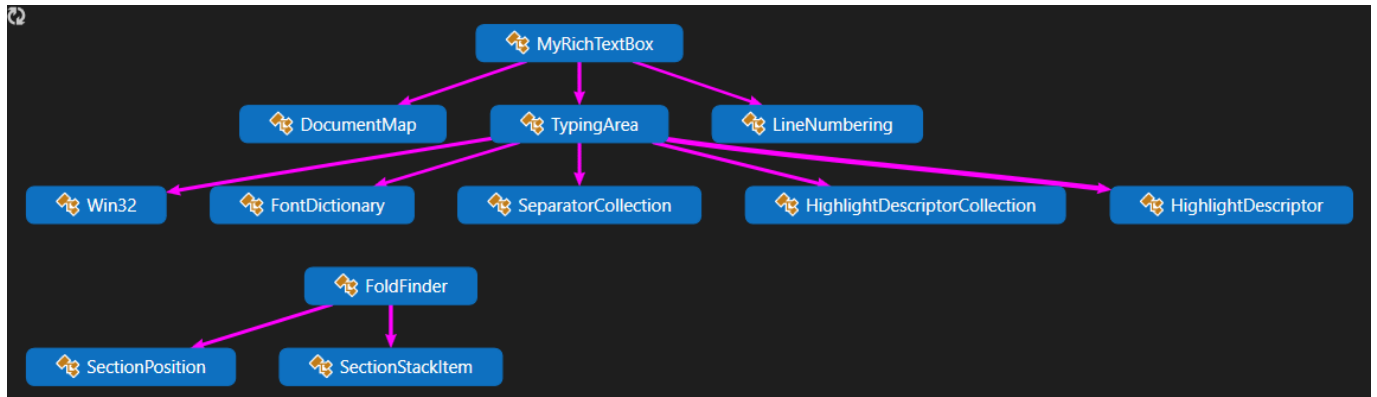
2.1. Các thành phần chính



Chương trình của nhóm thực hiện sẽ có 2 phần chính:

- GUI: bao gồm hiện thực hóa thiết kế giao diện và các nút xử lí, các tab page của chương trình.
- SyntaxHighlightingTextbox: Là một thư viện liên kết động (Dynamic-link library) được viết để giúp GUI thực hiện các chức năng quan trọng của một Code Editor.

2.2. Thư viện liên kết động *SyntaxHighlightingTextbox*



Trong sơ đồ trên, các class chính là :

- **MyRichTextBox:** kế thừa từ UserControl, đây là nơi tập hợp TypingArea, LineNumbering, DocumentMap để tạo thành một nơi thao tác soạn thảo đầy đủ cho người dùng.
- **DocumentMap:** Component kế thừa từ RichTextBox có chức năng xem tổng quan phần soạn thảo với kích thước thu nhỏ.
- **TypingArea:** Component kế thừa từ RichTextBox, là phần soạn thảo hỗ trợ các chức năng quan trọng như đổi màu các keywords trong các ngôn ngữ lập trình, tự gợi ý từ và hoàn thành từ, tự hoàn thành dấu đóng mở ngoặc, tìm kiếm và thay thế, ...
- **LineNumbering:** Component kế thừa từ TextBox để đánh số hàng tương ứng với phần soạn thảo (TypingArea).
- **FoldFinder:** Class hỗ trợ việc thu gọn code.
- **HighlightDescriptor:** Class dùng để lưu trữ các thông tin về cách để làm đổi màu một chữ như: Kiểu Font áp dụng, màu chữ áp dụng, từ khóa cụ thể, cách nhận dạng từ khóa, cách làm đổi màu từ khóa, ...

2.2.1. TypingArea

Các thuộc tính chính			
STT	Tên thuộc tính	Kiểu thuộc tính	Ý nghĩa
1	separators	SeparatorCollection	Là tập hợp các dấu phân cách. (Giữa 2 dấu phân cách sẽ là một token – dùng để so

			sánh với keywords có sẵn của ngôn ngữ).
2	descriptors	HighlightDescriptorCollection	Là tập hợp các mô tả cho việc highlight một keyword cụ thể.
3	requiresSaving	int	Biến dùng để nhận biết phần nhập liệu có thay đổi hay chưa, hỗ trợ chức năng Save.
4	UndoRedoInfo	struct	Là một struct được tạo ra để lưu trữ các thông tin cho việc Undo, Redo bao gồm nội dung văn bản, vị trí con trỏ, vị trí thanh cuộn.
5	undoStack	Stack<UndoRedoInfo>	Stack để lưu trữ thông tin Undo.
6	redoStack	Stack<UndoRedoInfo>	Stack để lưu trữ thông tin Redo.
7	caseSensitive	bool	Biến dùng để nhận biết ngôn ngữ đang soạn thảo có phân biệt chữ in hoa hay không.
8	enableHighlight	bool	Biến dùng để bật tắt chức năng đổi màu chữ.
9	autoCompleteListBox	ListBox	ListBox chứa các từ gợi ý với các ký tự vừa gõ vào phần soạn thảo.

Các phương thức chính			
STT	Tên phương thức	Kiểu thuộc tính	Chức năng
1	OnTextChanged(EventArgs e)	void	Event Handler được override để bắt sự kiện thay đổi chữ. Dùng cho việc nhận dạng và đổi mẫu các từ khớp với keyword có sẵn.
2	Undo()	void	Thực hiện chức năng Undo.
3	Redo()	void	Thực hiện chức năng Redo.

4	Highlight()	void	Làm đổi màu chữ bằng cách xây dựng lại Property Rtf của RichTextBox.
5	LoadKeywordsToListBox(string characters)	void	Đưa các từ bắt đầu bằng các kí tự {characters} vào ListBox.
6	AutoShowListBox()	void	Hiện ListBox tại vị trí đang gõ văn bản để gợi ý từ.
7	CompleteMatchedWord()	void	Paste chữ được chọn trong ListBox.
8	FindAll(string textToSearch)	List<int>	Tìm và trả về tất cả các từ chứa {textToSearch}.
9	BraceMatching(char input)	void	Tự động thêm dấu đóng của {input}.
10	AutoIndenting()	void	Tự động thụt đầu hàng tương ứng với hàng trước đó

Thông tin thêm:

- RichTextBox.Rtf: Là một property kiểu string chứa văn bản theo theo định dạng RTF (Rich Text Format). Sử dụng các tag đổi định dạng lại văn bản đã có.

Phần hiển thị

This line is the default color

This line is red

This line is the default color

Thuộc tính RTF

```
{\rtf1\ansi\deff0
{\colortbl;\red0\green0\blue0;\red255\green0\blue0;}
This line is the default color\line
\cf2
This line is red\line
\cf1
This line is the default color
}
```

2.2.2. HighlightDescriptor

Các thuộc tính chính			
STT	Tên thuộc tính	Kiểu thuộc tính	Ý nghĩa
1	color	Color	Màu chữ được áp dụng.

2	font	Font	Kiểu chữ được áp dụng.
3	token	string	Keyword cụ thể để so sánh.
4	closeToken	string	Dấu đóng ngoặc, nếu {token} là một dấu mở ngoặc.
5	highlightType	HighlightType	Kiểu áp dụng highlight.
6	descriptorRecognition	DescriptorRecognition	Kiểu nhận dạng các từ.
7	isUsedForAutoComplete	UsedForAutoComplete	Xác định dùng cho gợi ý từ hay không.

Thông tin thêm:

- HighlightType: là kiểu enumeration, thể hiện cách để highlight keyword. (ToEOL, ToEOW, ToCloseToken).
- DescriptorRecognition: là kiểu enumeration, thể hiện cách nhận dạng keyword. (IsNumber, StartsWith, WholeWord).
- UsedForAutoComplete: là kiểu enumeration (Yes, No).

2.2.3. LineNumbering

Các thuộc tính chính			
STT	Tên thuộc tính	Kiểu thuộc tính	Ý nghĩa
1	line	int	Tổng số dòng chữ nằm trong TypingArea.
2	pt	ioint	Xác định tọa độ.
3	First_Index	int	Xác định vị trí đầu tiên của kí tự trong TypingArea.
4	First_Line	int	Xác định dòng đầu tiên của TypingArea.
5	Last_Index	int	Xác định vị trí cuối cùng của kí tự trong TypingArea.
6	Last_Line	int	Xác định dòng cuối cùng của TypingArea.
7	SelectionAlignment	enum	Xác định vị trí của text trong khu vực chứa Line Number (căn giữa).

8	Width	int	Độ rộng của LineNumberingTextbox.
---	-------	-----	-----------------------------------

Các phương thức chính			
STT	Tên phương thức	Kiểu thuộc tính	Chức năng
1	AddLineNumbers()	void	Tự động thêm số vào LineNumberingTextbox sao cho ứng với các vị trí đã định sẵn.
2	OnTypingArea_TextChanged()	void	Xử lý sự kiện: cập nhật lại LineNumberingTextbox khi text trong TypingArea thay đổi.
3	OnTypingArea_VScroll()	void	Xử lý sự kiện: cập nhật lại LineNumberingTextbox khi người dùng lăn chuột lên xuống.
4	OnTypingArea_FontChanged()	void	Xử lý sự kiện: cập nhật kích thước của LineNumberingTextbox khi thay đổi font chữ trong TypingArea.
5	OnTypingArea_MouseDown()	void	Xử lý sự kiện: Khi click chuột lên TypingArea thì bỏ chọn nếu đang bôi đen tại LineNumberingTextbox.
6	OnTypingArea_SizeChanged()	void	Xử lý sự kiện: cập nhật kích thước của LineNumberingTextbox khi tăng giảm kích cỡ kí tự (Zoom in Zoom out).
7	getWidth()	void	Lấy độ rộng của TypingArea để cập nhật cho LineNumberingTextbox.

2.2.4. DocumentMap

Các thuộc tính chính			
STT	Tên thuộc tính	Kiểu thuộc tính	Ý nghĩa
1	Text	string	Kí tự trong DocumentMap sẽ giống y hệt như các kí tự trong TypingArea.

2	Rtf	string	Định dạng kí tự trong DocumentMap giống định dạng kí tự trong TypingArea.
3	Font	font	Font chữ trong DocumentMap sẽ giống font chữ trong TypingArea nhưng kích thước giảm đi.
4	GetScrollPos	int	Lấy vị trí thanh scroll của TypingArea.
5	nPos	int	Lưu vị trí của thanh scroll bên trong TypingArea.

Các phương thức chính			
STT	Tên phương thức	Kiểu thuộc tính	Chức năng
1	OnTypingArea_TextChanged()	void	Xử lý sự kiện: Cập nhật lại DocumentMap khi text trong TypingArea thay đổi.
2	OnTypingArea_Vscroll()	void	Xử lý sự kiện: Cập nhật lại vị trí thanh scroll trong DocumentMap để đồng bộ với vị trí thanh scroll trong TypingArea.
3	WndProc	void	Disable tất cả các sự kiện liên quan đến chuột trong DocumentMap

Thông tin thêm:

- WndProc: Vì DocumentMap chỉ là nơi để hiển thị tổng quan phần soạn thảo nên sẽ không có tương tác chuột với DocumentMap

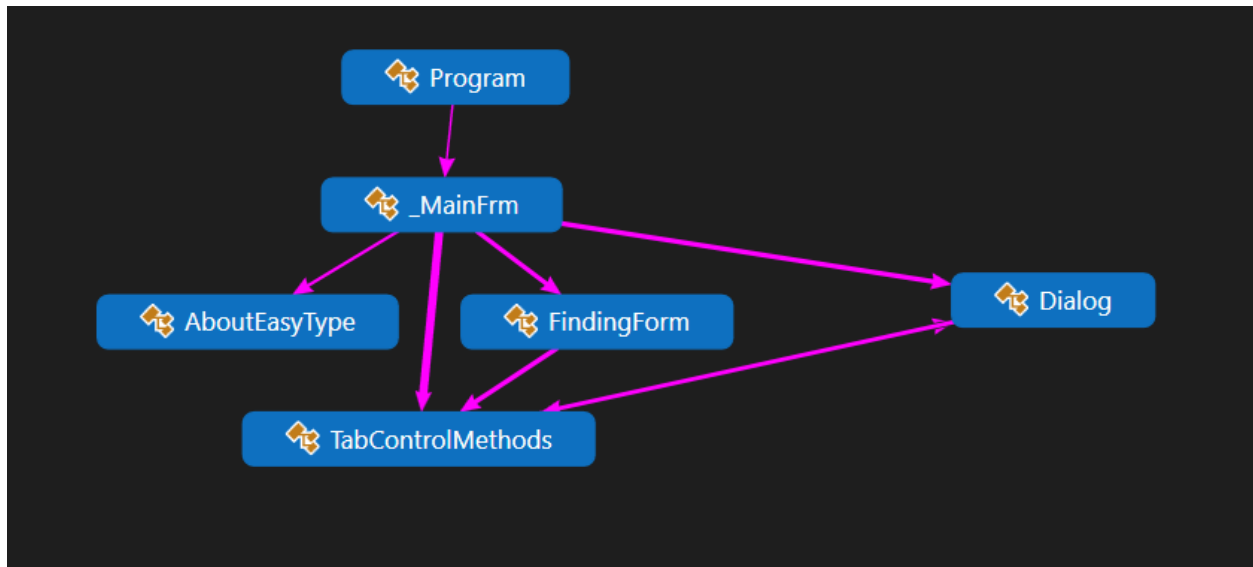
2.2.5. MyRichTextBox

Các thuộc tính chính			
STT	Tên thuộc tính	Kiểu thuộc tính	Ý nghĩa

1	lineNumberTextBox	LineNumbering	Kế thừa từ RichTextBox, giúp đánh số dòng ứng với kí tự trong TypingArea
2	typingArea	TypingArea	Kế thừa từ RichTextBox, nơi soạn thảo văn bản chính
3	documentMap	DocumentMap	Hiển thị tổng quan phần soạn thảo TypingArea với kích thước thu nhỏ lại
4	Font	Font	Font chữ được sử dụng trong MyRichTextBox, tức là toàn bộ TypingArea, LineNumbering và DocumentMap

Các phương thức chính			
STT	Tên phương thức	Kiểu thuộc tính	Chức năng
1	OnTypingArea_TextChanged()	void	Phân phối các sự kiện khi kí tự trong TypingArea thay đổi về cho LineNumberTextbox và DocumentMap
2	OnTypingArea_Vscroll()	void	Phân phối lại các sự kiện khi lăn con trỏ chuột về cho LineNumberTextbox và DocumentMap xử lí
3	OnTypingArea_FontChanged()	void	Phân phối sự kiện thay đổi font chữ trong TypingArea về cho LineNumberTextbox
4	OnTypingArea_MouseDown()	void	Phân phối xử lí sự kiện khi click chuột
5	OnTypingArea_SizeChanged()	void	Phân phối xử lí sự kiện khi thay đổi kích thước trong TypingArea

2.3. GUI



Trong phần GUI, ta có Form chính để chương trình thực thi đó là _MainFrm, chứa EventHandlers của các nút xử lý cũng như khởi tạo giao diện như đã thiết kế. Bên cạnh đó là 2 class hỗ trợ cho việc làm giao diện và 2 forms phụ sẽ được gọi khi có yêu cầu:

- AboutEasyType: Là form chứa Textbox đơn giản thể hiện thông tin về phần mềm, nhóm thức hiện, version.
- FindingForm: Là form để sử dụng cho chức năng tìm kiếm các từ người dùng nhập trong văn bản.
- TabControlMethods: Là một class quan trọng hỗ trợ việc thao tác với TabControl có sẵn của System.Windows.Forms.
- Dialog: Là một class phụ hỗ trợ việc xác nhận đóng tab, đóng chương trình.

TabControlMethods

Các thuộc tính chính			
STT	Tên thuộc tính	Kiểu thuộc tính	Ý nghĩa
1	tabControl	TabControl	Biến để lưu trữ TabControl đang thao tác với Windows.Forms.TabControl
Các phương thức chính			
STT	Tên phương thức	Kiểu thuộc tính	Chức năng

1	SetupTabControl(TabControl _tabControl)	void	Lưu trữ địa chỉ của Windows.Forms.TabControl đang hoạt động và thêm một số xử lý sự kiện để tùy biến.
2	TabControl_OnDrawItem(object sender, DrawItemEventArgs e)	void	EventHandler để tùy biết các Tab theo ý của nhóm.
3	TabControl_OnMouseDown(object sender, MouseEventArgs e)	void	EventHandler để hỗ trợ sự kiện đóng Tab khi click vào dấu x.
4	CreateNewTabPage(String tabName)	TabPage	Tạo tab mới với tên {tabName}.
5	CloseCurrentTabPage()	void	Đóng Tab hiện tại.
6	IsEmpty()	bool	Được tạo ra để tránh một số lỗi xảy ra khi các chức năng được yêu cầu nhưng không có TabPage nào để tương tác
7	TabControl_DragTab(object sender, DragEventArgs e)	void	EventHandler được gọi từ sự kiện DragOver, hỗ trợ việc kéo tab để đổi vị trí giữa các tabs.

IV. CÀI ĐẶT VÀ THỬ NGHIỆM

1. Môi trường và ngôn ngữ cài đặt

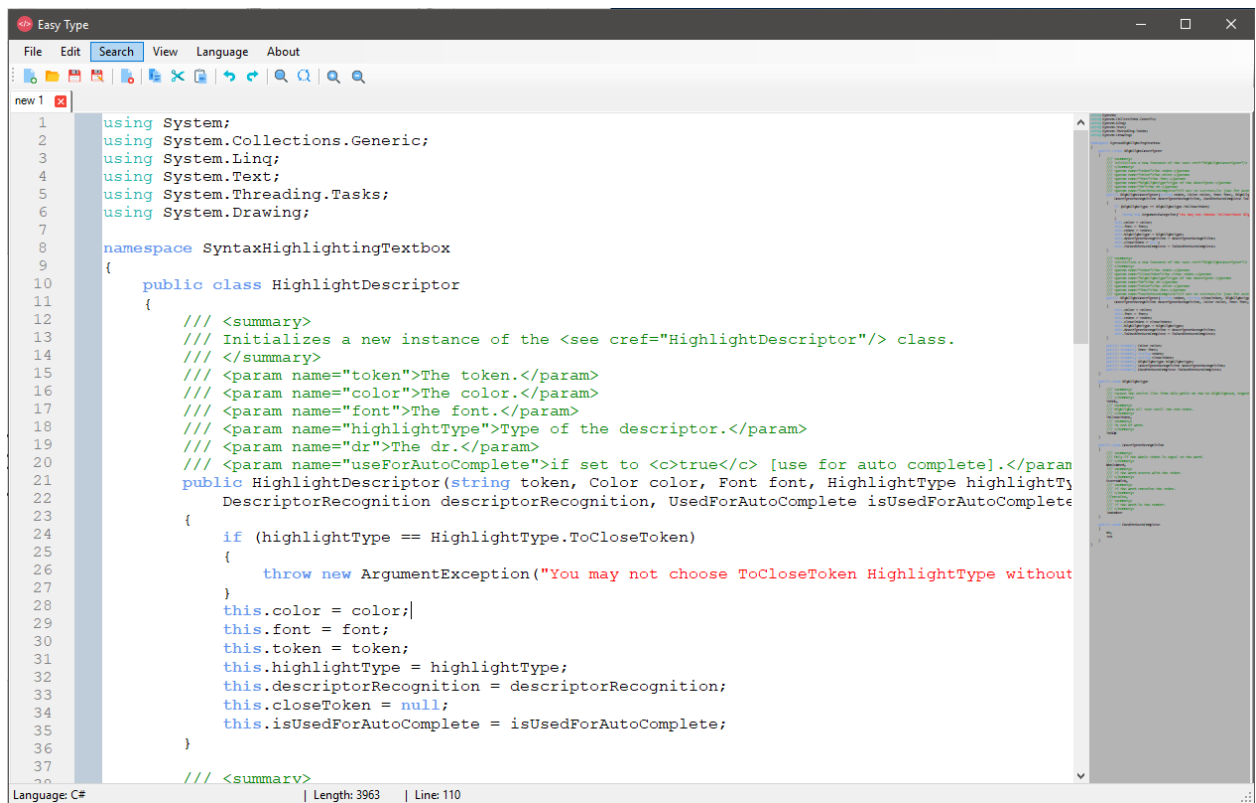
Chương trình được xây dựng trên ngôn ngữ C# và sử dụng .Net Framework 4.5 nên có thể chạy tốt trên tất cả các máy tính đã cài đặt Microsoft .NET Framework.

2. Thử nghiệm

Chạy thử chương trình với laptop cấu hình là:

- Processor: i7 – 3520M
- Ram: 4GB
- Hệ điều hành: Windows 10 Education.

Hình ảnh chạy và sử dụng thử của chương trình:



Trải nghiệm: Chương trình hoạt động khá tốt, có thể đưa vào sử dụng như một code editor thay thế Notepad++.

V. ĐÁNH GIÁ KẾT QUẢ

1. Kết quả đạt được

- Giao diện tab cơ bản.
- Hỗ trợ các ngôn ngữ C, C++, C#, VB, Javascript, SQL.
- Highlight các từ trong khi gõ với các từ khóa cố định.
- Tự động gợi ý và hoàn thành với từ khóa cố định.
- Document Map hoạt động tốt, tuy nhiên còn hạn chế.
- Tự động thụt hàng và tự động đóng ngoặc hỗ trợ thao tác soạn thảo hoạt động hiệu quả.
- Chương trình chỉ dừng ở mức chạy được, chưa thực sự tối ưu về hiệu năng.

2. Thuận lợi và khó khăn

a. Thuận lợi

- Kế thừa từ RichTextBox làm cho một số xử lý cơ bản trở nên dễ dàng.
- Đây là chủ đề không mới, nên nhiều nguồn tham khảo.
- Sử dụng winform nên dễ thiết kế, dễ cài đặt.

b. Khó khăn

- Xử lý highlight chưa có thuật toán tối ưu, còn chậm đối với đoạn text lớn.
- Nhóm chưa có kinh nghiệm làm một chương trình hoàn chỉnh nên thể tránh khỏi sai sót.

VI. KHÁC

1. Phân công công việc

Chức năng	Người phụ trách	Tình trạng
Syntax Highlight	Nghĩa, Tín	Hoàn thành
Brace Matching	Nghĩa	Hoàn thành
Auto Indenting	Nghĩa, Tiến	Hoàn thành
Tab Bar (Multi-Document)	Nghĩa	Hoàn thành
Document Map	Tín	Hoàn thành
Code Folding	Tín	Còn bị lỗi
Number Margin	Tín, Tiến	Hoàn thành
Tìm kiếm và thay thế	Tiến	Hoàn thành
GUI	Nghĩa, Tín, Tiến	Hoàn thành

2. Hướng phát triển phần mềm

- Nghiên cứu thêm để tối ưu các chức năng đã có của chương trình.
- Tự động thêm từ khóa để gợi ý và highlight khi người dùng gõ.
- Hỗ trợ compiler một số ngôn ngữ.
- Hỗ trợ thêm nhiều ngôn ngữ khác.

3. Tài liệu tham khảo

Gọi hàm SendMessage để xử lý một số thao tác với windows. Tham khảo từ MSDN:

[https://msdn.microsoft.com/en-us/library/windows/desktop/ms644950\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644950(v=vs.85).aspx)

Syntax highlighting trong RichTextBox. Tham khảo từ CodeProject:

<https://www.codeproject.com/Articles/10532/Syntax-highlighting-textbox-written-in-C>

Một số xử lý liên quan tới định dạng RTF. Tham khảo từ Pindari:

<http://www.pindari.com/rtf1.html>

Line number cho RichTextBox. Tham khảo từ CSharpCorner:

<https://www.c-sharpcorner.com/blogs/creating-line-numbers-for-richtextbox-in-c-sharp>

Và từ rất nhiều nguồn tài liệu khác.