Lời cảm ơn

Em xin được bày tỏ lòng biết ơn sâu sắc tới thầy giáo Đặng Quang Huy và thầy giáo Vũ Mạnh Khánh - giảng viên trường Đại học dân lập Hải Phòng đã tận tình hướng dẫn và tạo mọi điều kiện thuận lợi để em hoàn thành báo cáo thực tập tốt nghiệp của mình.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công nghệ thông tin - Trường Đại học dân lập Hải Phòng đã nhiệt tình giảng dạy và cung cấp những kiến thức quý báu để em có thể hoàn thành tốt đồ án tốt nghiệp này.

Cuối cùng, em xin cảm ơn gia đình và tất cả các bạn tập thể lớp CT1001 đã động viên, góp ý và trao đổi hỗ trợ cho em trong suốt thời gian vừa qua.

Em xin chân thành cảm ơn!

Hải Phòng, ngày.....tháng.....năm.2009.

Sinh viên

MỤC LỤC

LỜI MỞ ĐẦU

Xử lý ngôn ngữ tự nhiên (*natural language processing* - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp.

Cùng vói sự phát triển của khoa học máy tính, việc nghiên cứu xử lý ngôn ngữ tự nhiên hay cụ thể hơn là việc đưa xử lý tiếng nói và chữ viết vào máy tính đã và đang được đầu tư mạnh mẽ trên khắp thế giới trong đó có Việt Nam. Tuy đã đạt được nhiều thành tựu to lớn nhưng công việc này vẫn là ngành khoa học trắc trở và tiêu tốn nhiều công sức.

Chương 1 : GIỚI THIỆU VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

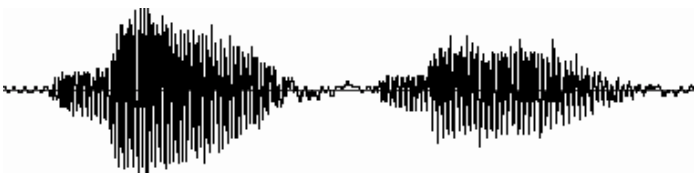
I. Tổng quan

Xử lý ngôn ngữ chính là xử lý thông tin khi đầu vào là "dữ liệu ngôn ngữ" (dữ liệu cần biến đổi), tức dữ liệu "văn bản" hay "tiếng nói". Các dữ liệu liên quan đến ngôn ngữ viết (văn bản) và nói (tiếng nói) đang dần trở nên kiểu dữ liệu chính con người có và lưu trữ dưới dạng điện tử. Đặc điểm chính của các kiểu dữ liệu này là không có cấu trúc hoặc nửa cấu trúc và chúng không thể lưu trữ trong các khuôn dạng cố định như các bảng biểu. Theo đánh giá của công ty Oracle, hiện có đến 80% dữ liệu không cấu trúc trong lượng dữ liệu của loài người đang có [Oracle Text]. Với sự ra đời và phổ biến của Internet, của sách báo điện tử, của máy tính cá nhân, của viễn thông, của thiết bị âm thanh, … người người ai cũng có thể tạo ra dữ liệu văn bản hay tiếng nói. Vấn đề là làm sao ta có thể xử lý chúng, tức chuyển chúng từ các dạng ta chưa hiểu được thành các dạng ta có thể hiểu và giải thích được, tức là ta có thể tìm ra thông tin, tri thức hữu ích cho mình.

Giả sử chúng ta có các câu sau trong các tiếng nước ngoài:

- "We meet here today to talk about Vietnamese language and speech processing."
- "Aujourd'hui nous nous réunissons ici pour discuter le traitement de langue et de parole vietnamienne."
- "Мы встрачаемся здесь сегодня, чтобы говорить о вьетнамском языке и обработке речи."

Nếu có ai đó dịch, hoặc có một chương trình máy tính dịch (biến đổi) chúng ra tiếng Việt, ta sẽ hiểu nghĩa các câu trên đều là: "Hôm nay chúng ta gặp nhau ở đây để bàn về xử lý ngôn ngữ và tiếng nói tiếng Việt."Nếu các câu này được lưu trữ như các tệp tiếng Anh, Pháp, Nga và Việt như ta nhìn thấy ở trên, ta có các dữ liệu "văn bản". Nếu ai đó đọc các câu này, ghi âm lại, ta có thể chuyển chúng vào máy tính dưới dạng các tệp các tín hiệu (signal) "tiếng nói". Tín hiệu sóng âm của hai âm tiết tiếng Việt có thể nhìn thấy như sau



Hình 1 : Tín hiệu sóng âm của hai âm tiêt Tiếng Việt

Tuy nhiên, một văn bản thật sự (một bài báo khoa học chẳng hạn) có thể có đến hàng nghìn câu, và ta không phải có một mà hàng triệu văn bản. Web là một nguồn dữ liệu văn bản khổng lồ, và cùng với các thư viện điện tử − khi trong một tương gần các sách báo xưa nay và các nguồn âm thanh được chuyển hết vào máy tính (chẳng hạn bằng các chương trình nhận dạng chữ, thu nhập âm thanh, hoặc gõ thẳng vào máy) − sẽ sớm chứa hầu như toàn bộ kiến thức của nhân loại. Vấn đề là làm sao "xử lý" (chuyển đổi) được khối dữ liệu văn bản và tiếng nói khổng lồ này qua dạng khác để mỗi người có được thông tin và tri thức cần thiết từ chúng.

II. Cơ sở khoa học

II.1 Một số khái niệm cơ bản

a. Ngôn ngữ tự nhiên

Ngôn ngữ là hệ thống để giao thiệp hay suy luận dùng một cách biểu diễn phép ẩn dụ và một loại ngữ pháp theo logic, mỗi cái đó bao hàm một tiêu chuẩn hay sự thật thuộc lịch sử và siêu việt. Nhiều ngôn ngữ sử dụng điệu bộ, âm thanh, lý hiệu, hay chữ viết, và cố gắng truyền khái niệm, ý nghĩa, và ý nghĩ, nhưng mà nhiều khi những khía cạnh này nằm sát quá, cho nên khó phân biệt nó.

b. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người.Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ - công cụ hoàn hảo nhất của tư duy và giao tiếp.

c. Trí tuệ nhân tạo

Trí tuệ nhân tạo hay trí thông minh nhân tạo (tiếng Anh: artificial intelligence hay machine intelligence, thường được viết tắt là AI) là trí tuệ được biểu diễn bởi bất cứ một hệ thống nhân tạo nào. Thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo.

d. Nhập nhằng

Nhập nhằng trong ngôn ngữ học là hiện tượng thường gặp, trong giao tiếp hàng ngày con người ít để ý đến nó bởi vì họ xử lý tốt hiện tượng này. Nhưng trong

các ứng dụng liên quan đến xử lý ngôn ngữ tự nhiên khi phải thao tác với ý nghĩa từ vựng mà điển hình là dịch tự động nhập nhằng trở thành vấn đề nghiêm trọng. Ví dụ trong một câu cần dịch có xuất hiện từ "đường" như trong câu "ra chợ mua cho mẹ ít đường" vấn đề nảy sinh là cần dịch từ này là road hay sugar, con người xác định chúng khá dễ dàng căn cứ vào văn cảnh và các dấu hiệu nhận biết khác nhưng với máy thì không. Một số hiện tượng nhập nhằng: Nhập nhằng ranh giới từ, Nhập nhằng từ đa nghĩa, Nhập nhằng từ đồng âm (đồng tự), Nhập nhằng từ loại.

II.2 Lý thuyết thông tin

a. Khái niệm

Lý thuyết thông tin nghiên cứu về: Áp dụng các công cụ toán học trong việc lượng hóa data cho mục đích lưu trữ và truyền dữ liệu. Độ đo thông tin là Entropy, là số lượng bít trung bình cần thiết để cho việc lưu trữ hay truyền dữ liệu. Đóng vai trò quan trọng trong xử lý thông tin bằng các phương pháp thống kê, đặc biệt trong NLP

b. Entropy

Entropy là một độ đo thông tin. Entropy ~ hỗn độn, mờ, trái nghĩa với order, ...

Đo độ không chắc chắn: Entropy thấp -> Đo độ không chắc chắn thấp ; Entropy cao -> Đo độ không chắc chắn cao. Trong vật lý: Entropy giảm khi năng lượng được sử dụng. Ký hiệu p(x) là một phân bố của một biến ngẫu nhiên X. Ω là không gian mẫu của X Entropy được tính như sau:

$H(X) = - \sum x \in \Omega \quad p(x) \log 2 p(x)$.

Đơn vị: bits (log10: nats) .

Kí hiệu: H(X) = Hp(X) = H(p)

c. Perplexity - Cross Entropy

c. 1. Entropy liên quan thế nào đến hiểu ngôn ngữ?

Liên quan đến sự ko chính xác: một vấn đề càng có nhiều thông tin thì Entropy càng thấp. Có nhiều mô hình -> entropy đo chất lượng của các mô hình?

Ví dụ: mô hình mã hóa ký tự với trung bình số bít sử dụng trên mỗi ký tự là 2.5 Đây là mô hình ngôn ngữ 0-gram, nếu đặt trong sự liên kết của các âm tiết thì chúng ta có thể sinh được mô hình tốt hơn, chẳng hạn cho entropy 1.22 bít trên một ký tự.

c. 2. Perplexity

Entropy của một phân bố p(X) là: Hp(X)Thì giá trị 2H được gọi là perplexity

perplexity là số lượng mẫu trung bình mà một biến phải lựa chọn.Perlexity càng bé (tức là entropy càng bé) thì mô hình càng tốt <=> số bít dùng để mã hóa thông tin càng bé.

Ví dụ: Cho 8 con ngựa với xác suất lựa chọn như sau:

Ngựa 1: 1/2 ngựa 2: 1/4 ngựa 3: 1/8 ngựa 4: 1/16

Ngựa 5: 1/64 ngựa 2: 1/64 ngựa 3: 1/64 ngựa 4: 1/64

c.3. Entropy rate

Tính entropy của một dãy các từ trong một ngôn ngữ L

H(w1,...,wn) = - ∑W∈L p(W1n)log(W1n)

Entropy rate được coi như per-word entropy. Coi một ngôn ngữ như một quá trình ngẫu nhiên sản xuất một dãy các từ. Cần quan tâm đến một dãy vô hạn từ. Entropy rate H(L) được định nghĩa như sau:

$$H(L) = \lim_{n->\infty} \frac{1}{n} H(w_1,..., w_n) = \lim_{n->\infty} -\frac{1}{n} \sum_{\in L} p(w_1,..., w_n) \log p(w_1,..., w_n)$$

c.4 . Cross Entropy

Cross entropy được sử dụng khi chúng ta không biết phân bố thật p.

Cross-entropy của phân bố m của phân bố thật p được định nghĩa:

$$H(p,m) = \lim_{n->\infty} -\frac{1}{n} \sum_{\in L} p(w_1,..., w_n) \log m(w_1,..., w_n) = \lim_{n->\infty} -\frac{1}{n} \log m(w_1,..., w_n)$$

(theo lý thuyết Shannon-McMillan-Breiman)

c.5. Cross entropy để so sánh các mô hình : H(p) ≤ H(p,m)

Cross entropy H(p,m) là cận trên của entropy H(p).

Mô hình m càng chính xác thì cross entropy H(p,m) càng gần với entropy H(p).

Độ khác nhau H(p,m) và H(p) đo độ chính xác của mô hình m.

c.6. Các công thức Cross Entropy

Cross entropy giữa biến X với phân bố xác suất đúng p(x) và một phân bố m được tính như sau:

$$H(X,m) = H(X) + D(p \parallel m) = -\sum_{x} p(x) \log m(x)$$

Chú ý: D(p||q) = ∑x p(x) log2 (p(x)/q(x))

II.3 Quy trình xử lý ngôn ngữ tự nhiên

Để máy tính có thể hiểu và thực thi một chương trình được viết bằng ngôn ngữ cấp cao, ta cần phải có một trình biên dịch thực hiện việc chuyển đổi chương

trình đó sang chương trình ở dạng ngôn ngữ đích. Chương này trình bày một cách tổng quan về cấu trúc của một trình biên dịch và mối liên hệ giữa nó với các thành phần khác - "họ hàng" của nó - như bộ tiền xử lý, bộ tải và soạn thảo liên kết, v.v. Cấu trúc của trình biên dịch được mô tả trong chương là một cấu trúc mức quan niệm bao gồm các giai đoạn: Phân tích từ vựng, Phân tích cú pháp, Phân tích ngữ nghĩa, Sinh mã trung gian, Tối ưu mã và Sinh mã đích. Nói một cách đơn giản, trình biên dịch là một chương trình làm nhiệm vụ đọc một chương trình được viết bằng một ngôn ngữ - ngôn ngữ nguồn (source language) - rồi dịch nó thành một chương trình tương đương ở một ngôn ngữ khác - ngôn ngữ đích (target languague). Một phần quan trọng trong quá trình dịch là ghi nhận lại các lỗi có trong chương trình nguồn để thông báo lại cho người viết chương trình.

Hình 2 : Một trình biên dịch

a. Phân tích từ vựng (Lexical Analysis)

Trong một trình biên dịch, giai đoạn phân tích từ vựng sẽ đọc chương trình nguồn từ trái sang phải (quét nguyên liệu - scanning) để tách ra thành các thẻ từ (token).

Ví dụ 1.2:

Quá trình phân tích từ vựng cho câu lệnh gán position := initial + rate * 60 sẽ tách thành các token như sau:

1. Danh biểu position
2. Ký hiệu phép gán :=
3. Danh biểu initial
4. Ký hiệu phép cộng (+)
5. Danh biểu rate
6. Ký hiệu phép nhân (*)
7. Số 60

Trong quá trình phân tích từ vựng các khoảng trắng (blank) sẽ bị bỏ qua.

b. Phân tích cú pháp (Syntax Analysis)

Giai đoạn phân tích cú pháp thực hiện công việc nhóm các thẻ từ của chương trình nguồn thành các ngữ đoạn văn phạm (grammatical phrase), mà sau đó sẽ được trình biên dịch tổng hợp ra thành phẩm. Thông thường, các ngữ đoạn văn

phạm này được biểu diễn bằng dạng cây phân tích cú pháp (parse tree) với :

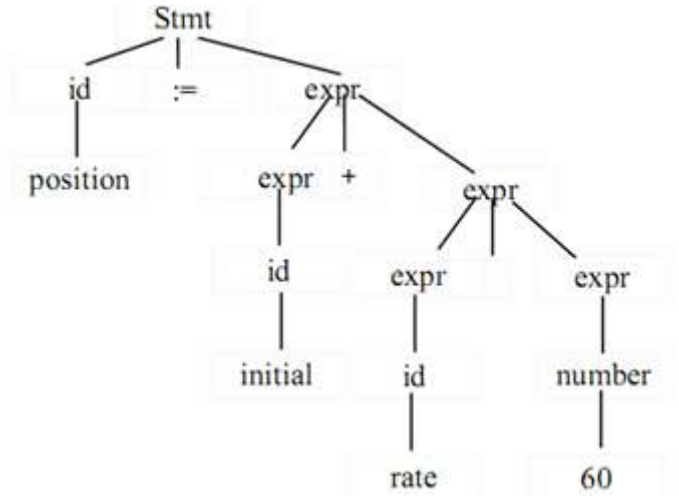
- Ngôn ngữ được đặc tả bởi các luật sinh.
- Phân tích cú pháp dựa vào luật sinh để xây dựng cây phân tích cú pháp.

Ví dụ 1.3: Giả sử ngôn ngữ đặc tả bởi các luật sinh sau :

Stmt → id := expr

expr → expr + expr | expr * expr | id | number

Với câu nhập: position := initial + rate * 60, cây phân tích cú pháp được xây dựng như sau :



Hình 3 :Một cây phân tích cú pháp

Cấu trúc phân cấp của một chương trình thường được diễn tả bởi quy luật đệ qui.

Ví dụ 1.4:

1) Danh biểu (identifier) là một biểu thức (expr).

2) Số (number) là một biểu thức.

3) Nếu expr1 và expr2 là các biểu thức thì:

expr1 + expr2

expr1 * expr2

(expr)

4) Cũng là những biểu thức. Câu lệnh (statement) cũng có thể định nghĩa đệ qui:

4.1) Nếu id1 là một danh biểu và expr2 là một biểu thức thì id1 := expr2 là một lệnh (stmt).

4.2) Nếu expr1 là một biểu thức và stmt2 là một lệnh thì while (expr1) do stmt2 và if (expr1) then stmt2: đều là các lệnh. Người ta dùng các qui tắc đệ qui như trên để đặc tả luật sinh (production) cho ngôn ngữ. Sự phân chia giữa quá trình

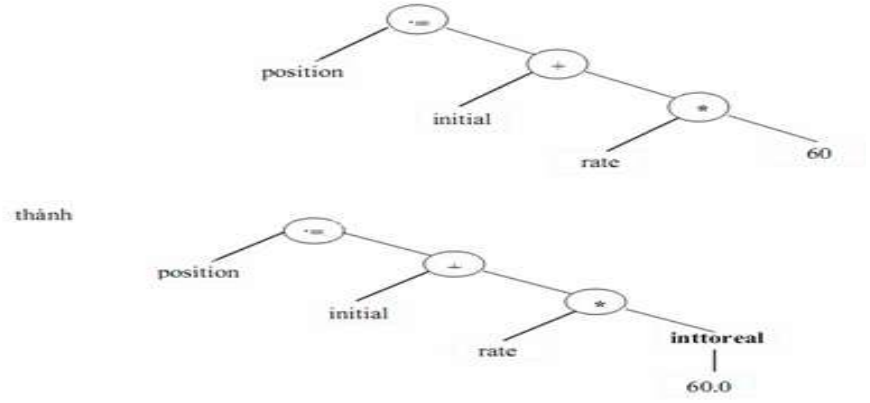
phân tích từ vựng và phân tích cú pháp cũng tuỳ theo công việc thực hiện.

c. Phân tích ngữ nghĩa (Semantic Analysis)

Giai đoạn phân tích ngữ nghĩa sẽ thực hiện việc kiểm tra xem chương trình nguồn có chứa lỗi về ngữ nghĩa hay không và tập hợp thông tin về kiểu cho giai đoạn sinh mã về sau. Một phần quan trọng trong giai đoạn phân tích ngữ nghĩa là kiểm tra kiểu (type checking) và ép chuyển đổi kiểu.

Ví dụ 1.5: Trong biểu thức position := initial + rate * 60

Các danh biểu (tên biến) được khai báo là real, 60 là số integer vì vậy trình biên dịch đổi số nguyên 60 thành số thực 60.0

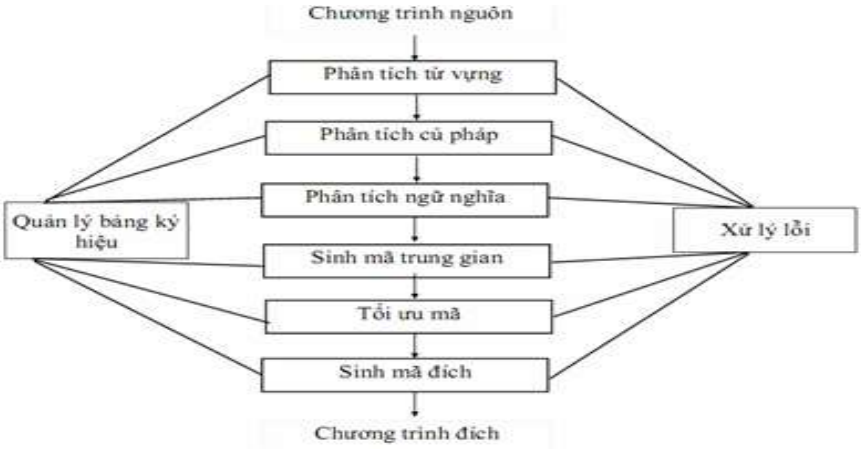


Hình 4: Chuyển đổi kiểu trên cây phân tích cú pháp

d. Các giai đoạn của trình biên dịch

Một trình biên dịch được chia thành các giai đoạn, mỗi giai đoạn chuyển chương trình nguồn từ một dạng biểu diễn này sang một dạng biểu diễn khác.

VÍ DỤ: Một cách phân rã điển hình trình biên dịch được trình bày trong hình



:

Hình 5: Các giai đoạn của một trình biên dịch

Việc quản lý bảng ký hiệu và xử lý lỗi được thực hiện xuyên suốt qua tất cả

các giai đoạn. Các giai đoạn mà chúng ta đề cập ở trên là thực hiện theo trình tự logic của một trình biên dịch. Nhưng trong thực tế, cài đặt các hoạt động của nhiều hơn một giai đoạn có thể được nhóm lại với nhau. Thông thường chúng được nhóm thành hai nhóm cơ bản, gọi là: kỳ đầu (Front end) và kỳ sau (Back end).

1. Kỳ đầu (Front End)

Kỳ đầu bao gồm các giai đoạn hoặc các phần giai đoạn phụ thuộc nhiều vào ngôn ngữ nguồn và hầu như độc lập với máy đích. Thông thường, nó chứa các giai đoạn sau: Phân tích từ vựng, Phân tích cú pháp, Phân tích ngữ nghĩa và Sinh mã trung gian. Một phần của công việc tối ưu hóa mã cũng được thực hiện ở kỳ đầu. Front end cũng bao gồm cả việc xử lý lỗi xuất hiện trong từng giai đoạn.

2. Kỳ sau (Back End)

Kỳ sau bao gồm một số phần nào đó của trình biên dịch phụ thuộc vào máy đích và nói chung các phần này không phụ thuộc vào ngôn ngữ nguồn mà là ngôn ngữ trung gian. Trong kỳ sau, chúng ta gặp một số vấn đề tối ưu hoá mã, phát sinh mã đích cùng với việc xử lý lỗi và các thao tác trên bảng ký hiệu.

II.4 Một số thuật toán phân tích cú pháp

1. Topdown

Phân tích từ trên xuống, từ trái qua phải.

Khi gặp một từ (terminal) thì phân tích nút tiếp theo.

Khi không tương ứng với input word thì quay lui.

2. Bottom-up

Là một dạng của shift - reduce actions.

Khi gặp vế phải của một luật thì thu gọn thành vế trái.

Khi không phân tích được tiếp thì quay lui.

3. CYK (Cocke-Younger-Kasami)

Văn phạm dạng chuẩn Chomsky (Chomsky Normal Form).

Các luật thuộc một trong 2 dạng:

A -> B C

A -> a

Ví dụ:

S -> X Y

X -> X A | a | b

Y -> A Y | a

A -> a

Phân tích câu “babaa” -> không sinh ra câu.

“baaa” -> sinh ra câu.

S, X			
S, X	S,Y		
S, X	S, X, Y	S, X, Y	
X	X, Y, A	X, Y, A	X, Y, A
b	a	a	a

```
S -> X Y
X -> X A | a | b
Y -> A Y | a
A -> a
"baaa"
```

Xác định các đặc điểm sau đây:

1) Sinh ra giá trị một nút như thế nào?

A[i,j] <- ? + ?

2) Lưu lại đường đi như thế nào để sinh lại cây.

Tính nhập nhằng: một A[,] có thể có nhiều tag, mỗi tag lại được dẫn xuất bằng nhiều cách.

3) Tại sao thuật toán CYK lại cần văn phạm dạng chuẩn Chomsky.

Phân tích câu:

“book that flight”

“book the flight through Houston”

$S \rightarrow NP\ VP$	$Det \rightarrow that \mid this \mid a$
$S \rightarrow Aux\ NP\ VP$	$Noun \rightarrow book \mid flight \mid meal \mid money$
$S \rightarrow VP$	$Verb \rightarrow book \mid include \mid prefer$
$NP \rightarrow Pronoun$	$Pronoun \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$Proper\text{-}Noun \rightarrow Houston \mid TWA$
$NP \rightarrow Det\ Nominal$	$Aux \rightarrow does$
$Nominal \rightarrow Noun$	$Preposition \rightarrow from \mid to \mid on \mid near \mid through$
$Nominal \rightarrow Nominal\ Noun$	
$Nominal \rightarrow Nominal\ PP$	
$VP \rightarrow Verb$	
$VP \rightarrow Verb\ NP$	
$VP \rightarrow Verb\ NP\ PP$	
$VP \rightarrow Verb\ PP$	
$VP \rightarrow VP\ PP$	
$PP \rightarrow Preposition\ NP$	

Figure 13.1 The $\mathscr{L}_1$ miniature English grammar and lexicon.

Chuyển từ văn phạm CFG sang văn phạm dạng chuẩn Chomsky.

1) A -> B C D

   A -> X D

   X -> B C

2) Bỏ luật dạng A -> B

   Với mọi B -> α, sinh luật A -> α

$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

Figure 13.8 $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

S						
	VP					
S						
	VP			PP		
S		NP			NP	
NP	V, VP	Det	N	P	Det	N
she	eats	a	fish	with	a	fork

Thử sinh ra một văn phạm tương ứng.

4. Thuật toán parsing CYK

```
function CKY-PARSE(words, grammar) returns table

  for j ← from 1 to LENGTH(words) do
      table[j − 1, j] ← {A | A → words[j] ∈ grammar }
      for i ← from j − 2 downto 0 do
          for k ← i + 1 to j − 1 do
              table[i,j] ← table[i,j] ∪
                           {A | A → BC ∈ grammar,
                                B ∈ table[i,k],
                                C ∈ table[k, j] }
```

Figure 13.10 The CKY algorithm

Đặc điểm:

- Có thể chuyển mọi văn phạm dạng CFG về dạng chuẩn Chomsky.
- Searching theo kiểu Bottom-up.
- Độ phức tạp phân tích là O(n3).
- Thuật toán là một dạng của dynamic programming.
- Có thể mở rộng thuật toán CYK để phân tích văn phạm xác suất.

III. Các ứng dụng của xử lý ngôn ngữ tự nhiên

1. Nhận dạng tiếng nói (speech recognition): Từ sóng tiếng nói, nhận biết và chuyển chúng thành dữ liệu văn bản tương ứng. Giúp thao tác của con người trên các thiết bị nhanh hơn và đơn giản hơn, chẳng hạn thay vì gõ một tài liệu nào đó bạn đọc nó lên và trình soạn thảo sẽ tự ghi nó ra. Đây cũng là bước đầu tiên cần phải thực hiện trong ước mơ thực hiện giao tiếp giữa con người với robot. Nhận

dạng tiếng nói có khả năng trợ giúp người khiếm thị rất nhiều.

2. Tổng hợp tiếng nói (speech synthesis): Từ dữ liệu văn bản, phân tích và chuyển thành tiếng người nói. Thay vì phải tự đọc một cuốn sách hay nội dung một trang web, nó tự động đọc cho chúng ta. Giống như nhận dạng tiếng nói, tổng hợp tiếng nói là sự trợ giúp tốt cho người khiếm thị, nhưng ngược lại nó là bước cuối cùng trong giao tiếp giữa người với robot.

3. Nhận dạng chữ viết (optical character recognition, OCR): Từ một văn bản in trên giấy, nhận biết từng chữ cái và chuyển chúng thành một tệp văn bản trên máy tính. có hai kiểu nhận dạng: Thứ nhất là nhận dạng chữ in như nhận dạng chữ trên sách giáo khoa rồi chuyển nó thành dạng văn bản điện tử như dưới định dạng doc của Microsoft Word chẳng hạn.Phức tạp hơn là nhận dạng chữ viết tay, có khó khăn bởi vì chữ viết tay không có khuôn dạng rõ ràng thay đổi từ người này sang người khác. Với chương trình nhận dạng chữ viết in có thể chuyển hàng ngàn đầu sách trong thư viện thành văn bản điện tử trong thời gian ngắn. Nhận dạng chữ viết của con người có ứng dụng trong khoa học hình sự và bảo mật thông tin (nhận dạng chữ ký điện tử).

4. Dịch tự động (machine translation): Từ một tệp dữ liệu văn bản trong một ngôn ngữ (tiếng Anh chẳng hạn), máy tính dịch và chuyển thành một tệp văn bản trong một ngôn ngữ khác. Một phần mềm điển hình về tiếng Việt của chương trình này là evtrans của Softex, dịch tự động từ tiếng Anh sang tiếng Việt và ngược lại, phần mềm từng được trang web vdict.com mua bản quyền, đây cũng là trang đầu tiên đưa ứng dụng này lên mạng. Có hai công ty tham gia vào lĩnh vực này cho ngôn ngữ tiếng Việt là công ty Lạc Việt (công ty phát hành từ điển Lạc Việt) và Google

5. Tóm tắt văn bản (text summarization): Từ một văn bản dài (mười trang chẳng hạn) máy tóm tắt thành một văn bản ngắn hơn (một trang) với những nội dung cơ bản.

6. Tìm kiếm thông tin (information retrieval): Từ một nguồn rất nhiều tệp văn bản hay tiếng nói, tìm ra những tệp có nội dung liên quan đến một vấn đề (câu hỏi) ta cần biết (hay trả lời)… Điển hình của công nghệ này là Google, một hệ tìm kiếm thông tin trên Web, mà hầu như chúng ta đều dùng thường xuyên. Cần nói

thêm rằng mặc dù hữu hiệu hàng đầu như vậy, Google mới có khả năng cho chúng ta tìm kiếm câu hỏi dưới dạng các từ khóa (keywords) và luôn "tìm" cho chúng ta rất nhiều tài liệu không liên quan, cũng như rất nhiều tài liệu liên quan đã tồn tại thì Google lại tìm không ra.

7. Trích chọn thông tin (information extraction): Từ một nguồn rất nhiều tệp văn bản hay tiếng nói, tìm ra những đoạn bên trong một số tệp liên quan đến một vấn đề (câu hỏi) ta cần biết hay trả lời. Một hệ trích chọn thông tin có thể "lần" vào từng trang Web liên quan, phân tích bên trong và trích ra các thông tin cần thiết, nói gọn trong tiếng Anh để phân biệt với tìm kiếm thông tin là "find things but not pages".

8. Phát hiện tri thức và khai phá dữ liệu văn bản (knowledge discovery and text data mining): Từ những nguồn rất nhiều văn bản thậm chí hầu như không có quan hệ với nhau, tìm ra được những tri thức trước đấy chưa ai biết. Đây là một vấn đề rất phức tạp và đang ở giai đoạn đầu của các nghiên cứu trên thế giới.

Có thể phân loại các bài toán:

1-3 thuộc lĩnh vực xử lý tiếng nói và xử lý ảnh (speech and image processing).

4-5 thuộc lĩnh vực xử lý văn bản (text processing).

6-8 thuộc lĩnh vực khai phá văn bản và Web (text and Web mining).

Chương 2: NGỮ PHÁP TIẾNG ANH

I. Các thì trong tiếng anh:

- Trong tiếng anh có 12 thì chính, được chia theo điều kiện thời gian như sau:

+ Hiện tại(Present):

- Đơn giản(Simple)
- Tiếp diễn(continuous)
- Hoàn thành(perfect)
- Hoàn thành tiếp diễn(perfect continuous)

+ Quá khứ(Past):

- Đơn giản(Simple)
- Tiếp diễn(continuous)
- Hoàn thành(perfect)
- Hoàn thành tiếp diễn(perfect continuous)

+ Tương lai(Future):

- Đơn giản(Simple)
- Tiếp diễn(continuous)
- Hoàn thành(perfect)
- Hoàn thành tiếp diễn(perfect continuous)

II: Cách sử dụng một số thì:

1. Thì hiện tại đơn(The Simple Present Tense):

1.1- Hình thức(Formation)

a. Thể khẳng định(Affirmative form)

S+ V…(Trong đó S là chủ ngữ, V là động từ thường)

* Nếu chủ ngữ là ngôi thứ 3 số ít(He,She, It, hoặc là một danh từ) thì động từ phải thêm "S" hoặc "ES"

b. Thể phủ định(Negative form)

S + do not / does not + V…

* "Does not" được sử dụng khi chủ ngữ là ngôi thứ 3 số ít, khi đó động từ ở dạng nguyên thể(không thêm "S" hoặc "ES").

c. thể nghi vấn(Interrogative form)

Do/Does + s + v…?

**Câu trả lời ngắn:*

+ Khẳng định: Yes, S + do/does

+ Phủ định: No, S + don't/doesn't

1.2 Cách sử dụng (The uasges)

a. Diễn tả một sự thật hiển nhiên

Ex: The earth goes round the sun.

b. Một hành động xảy ra hàng ngày, có tính lặp đi lặp lại

Ex: We go to school every day.

c. Diễn tả một hành động ở tương lai(thường dùng với các động từ chỉ sự chuyển động như: arrive, leave, return…)

Ex: She leaves tomorrow.

2. Thì hiện tại tiếp diễn(The present continuous/progressive tense)

2.1 Hình thức(formation)

a. Thể khẳng định(Affifmative form)

S + am/is/are + V_ing…

b. Thể phủ định(Negative form)

S + am not/ is not/ are not + V_ing…

Am not = *'m not, is not = isn't, are not = aren't.

c. Thể nghi vấn(Interrogative form)

Am/Is/Are + S + V_ing…?

**Câu trả lời ngắn:*

+Phẳng định: Yes, S + am/is/are

+Phủ định: No, S + 'm not/isn't/aren't

2.2 Cách sử dụng(The usages)

a. Diễn tả một hành động đang sảy ra tại thời điểm nói.

Ex: We are learning English now.

b. Một hành động xảy ra ở tương lai gần.

Ex: He is watching television tonight.

c. Một hành động được lặp đi lặp lại nhiều lần, gây bực mình(Thường có trạng từ "always")

Ex: That student is always making noise.

3. Thì hiện tại hoàn thành(The Present Prefect Tense)

3.1 Hình thức(Formation)

a. Thể khẳng định(Affirmative form)

S + have/has + PP… (PP : Quá khứ phân từ)

Have = 've, has = 's

* Nếu chủ ngữ lạ ngôi thứ 3 số ít thì chúng ta dùng "has".

b. Thể phủ định(Negative form)

S + haven't/ hasn't + PP…

c. Thể nghi vấn(Interrogative form)

Have/has + S + PP…?

**Câu trả lời ngắn:*

+Khẳng định: Yes, S + have/has

+Phủ định: No, S + haven't/hasn't

3.2 Cách sử dụng(The usages)

a. Diễn tả một hành động vừa mới xảy ra. Thường có trạng từ "just"

Ex: I have just bought this car.

b. Diễn tả một hành động xảy ra trong quá khứ không xác định thời gian. Thường có trạng từ "Already"

Ex: He has already read that book.

c. Diễn tả một hành động bắt đầu ở quá khứ và vẫn còn tiếp tục ở hiện tại.

Các trạng từ chỉ thời gian thường được dùng: ever, never, so far, since(điểm thời gian), for(khoảng thời gian)…

Ex: I have never driven a car. They have lived here since 1998.

4. Thì hiện tại hoàn thành tiếp diễn(The Present Prefect continuousTense)

4.1 Hình thức(Formation)

a. Thể khẳng định(Affirmative form)

S + have/has + been + V_ing…

b. Thể phủ định(Negative form)

S + haven't/ hasn't + Been + V_ing…

c. Thể nghi vấn(Interrogative form)

Have/has + S + Been + V_ing?

**Câu trả lời ngắn:*

+ Khẳng định: Yes, S + have/has

+ Phủ định: No, S + haven't/hasn't

4.2 Cách sử dụng(The usages)

a. Diễn tả một hành động bắt đầu còn liên tục đến hiện tại, chấm dứt ở hiện tại hoặc có thể kéo dài đến tương lai.

Ex: I have been waiting for you for a long time.

b. Lý do xảy ra ngay khi nói.

Ex: Your eyes are very red. Have you been crying?

5. Thì quá khứ đơn(The Simple Past Tense)

5.1 Hình thức(Formation)

a. Thể khẳng định(Affirmative form)

S + V_ed/V2…

* Nếu là động từ có quy tắc thì chúng ta thêm "ED" vào sau động từ thường, nếu là động từ bất quy tắc thì chúng ta sử dụng động từ ở cột 2 trong bảng động từ bất quy tắc.

b. Thể phủ định(Negative form)

S + did not + V…

did not = didn't

* Khi có trợ động từ "didn't" thì động từ theo sau trở về nguyên thể.

c. Thể nghi vấn(Interrogative form)

Did + S + V…?

* Khi có trợ động từ "Did" thi động từ ở dạng nguyên thể.

**Câu trả lời ngắn:*

+ Khẳng định: Yes, S + did

+ Phủ định: No, S + didn't

5.2. Cách sử dụng(The usages)

a. Diễn tả một hành động xảy ra tại một thời điểm xác định trong quá khứ và đã chấm dứt.

Ex: He stayed at home last night.

b. Diễn tả thói quen trong quá khứ.

Ex: She often played badminton when she was young.

c. Diễn tả các hành động xảy ra kế tiếp nhau trong quá khứ.

Ex: She came in, sat down and said nothing.

6. Thì quá khứ tiếp diễn (The Past continuous Tense)

6.1 Hình thức(Formation)

a. Thể khẳng định(Affirmative form)

S + was/were + V_ing…

Was: dùng cho ngôi I và ngôi thứ 3 số ít.

b. Thể phủ định(Negative form)

S + was not/ were not + V_ing…

Was not = wasn't, were not = weren't

c. Thể nghi vấn(Interrogative form)

Was/were + S + V_ing…?

**Câu trả lời ngắn:*

+ Khẳng định: Yes, S + was/were

+ Phủ định: No, S + wasn't/weren't

6.2 Cách sử dụng (The usages)

a. Diễn tả một hành động đang diễn ra tại một thời điểm trong quá khứ.

Ex: I was reading book at 8 o'clock last night.

b. Diễn tả một hành động đang xảy ra ở quá khứ thì bị một hành động khác cắt ngang. Hành động cắt ngang dùng ở thì quá khứ đơn.

Ex: We were watching TV when the light went out.

c. Một sự việc xảy ra và liên tục trong quá khứ.

Ex: I was sleeping all day yesterdat.

d. Chỉ 2 hành động xảy ra song song nhau trong quá khứ.

Ex:My father was reading newspaper while my mother was listening to music.

7. Thì tương lai đơn(The Simple Future Tense)

7.1 Hình thức(Formation)

a. Thể khẳng định(Affirmative form)

S + will/shall + V …

* Shall được dùng cho ngôi I và We. Trong văn nói và trong tiếng anh ngày nay người ta sử dụng "will" cho tất cả các ngôi.

'll: viết tắt của Shall và Will.

b. Thể phủ định(Negative form)

S + will not/ shall not + V…

c. Thể nghi vấn(Interrogative form)

Will/Shall + S + V…?

**Câu trả lời ngắn:*

+ Khẳng định: Yes, S + will/shall

+ Phủ định: No, S + won't/shan't

7.2 Cách sử dụng (The usages)

a. Diến tả một hành động sẽ xảy ra tại một thời điểm nào đó trong tương lai.

Ex: She'll be 20 on next Thursaday.

b. Diễn tả thói quen trong tương lai.

Ex: He will go for a walk after dinner.

c. Diễn tả một việc sẽ quyết định làm ngay lúc nói.

Ex: What would you like to drink? I'll have a mineral water.

Chương 3: GIỚI THIỆU NGÔN NGỮ VB 6.0

1. Giới thiệu

Visual Basic 6.0 (VB) là một ngôn ngữ lập trình hướng đối tượng, trực quan trên môi trường Windows. VB cung cấp một bộ công cụ hoàn chỉnh để đơn giản hóa việc triển khai lập trình ứng dụng, có thể nói đây là cách nhanh và tốt nhất để học và lập trình ứng dụng trên Microsoft Windows.

Phần "Visual- Trực quan" đề cập đến phương pháp được sử dụng để tạo giao diện đồ họa người dùng (GUI - Graphical User Interface). VB có sẵn rất nhiều những bộ phận trực quan gọi là các điều khiển (Controls) mà người lập trình có thể sắp đặt vị trí và quyết định các đặc tính của chúng trên một khung giao diện màn hình, gọi là form. Việc thiết kế các giao diện người dùng ứng dụng trên VB có thể hình dung đơn giản như việc vẽ giao diện trên Word hoặc trên Paint Prush của Windows.

Phần "Basic" đề cập đến ngôn ngữ BASIC (Beginners All-Purpose Symbolic Instruction Code), một ngôn ngữ lập trình đơn giản, dễ học, được viết ra cho các khoa học gia - những người không có thì giờ để học lập trình điện toán sử dụng. Tuy nhiên, ngôn ngữ Basic trong VB đã được cải thiện rất nhiều để phù hợp với phong cách lập trình hiện đại.

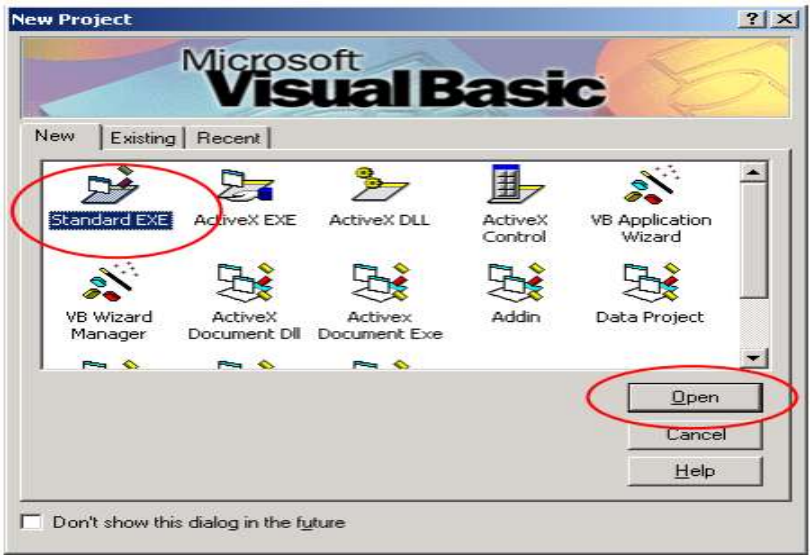
2. Các thao tác cơ bản trong VB

a. Khởi động

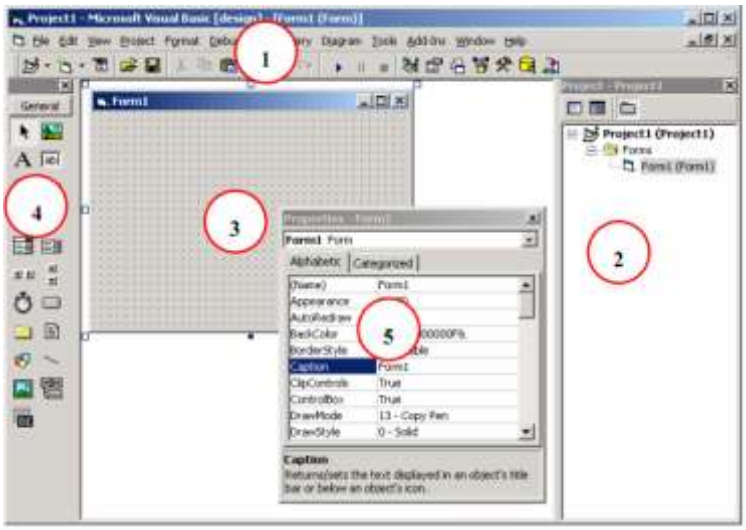
Sau khi gài đặt VB, có thể khởi động từ thanh tác vụ của Windows như sau:

Start | Programs | Microsoft Visual Studio 6.0 | Microsoft Visual Basic 6.0

Hộp thoại đầu tiên của phần mềm xuất hiện:



Để bắt đầu một ứng dụng mới, từ thẻ **New**, chọn **Standard EXE**, nhấn **Open** Môi trường làm việc VB xuất hiện:



Có rất nhiều các thành phần trong môi trường làm việc của VB. Ở mức đơn giản nhất có 5 thành phần được khoanh tròn trong hình trên đó là:

(1). Thanh thực đơn và thanh công cụ chuẩn của VB.

(2). Cửa sổ Project Explorer – nơi quản lý toàn bộ các thành phần mà người lập trình đã làm được trên dự án của VB hiện thời. Làm việc trên VB là làm việc trên các dự án (Projects). Mỗi dự án cần phải tạo ra nhiều thành phần để cấu thành như: giao diện, biểu mẫu báo cáo, thư viện,... tất cả những thành phần này sẽ đuợc quản lý trên cửa sổ Project Explorer.

(3). Biểu mẫu Form – nơi thường dùng để thiết kế các hộp thoại, cửa sổ - giao diện của người sử dụng với ứng dụng phần mềm.

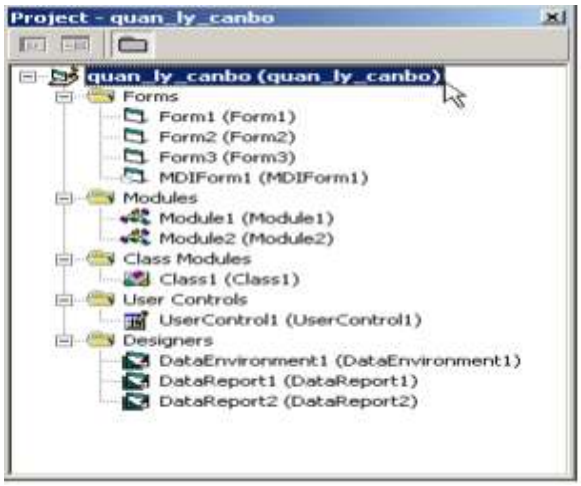
(4). Thanh công cụ ToolBox- nơi chứa các điều khiển (Control) giúp người lập trình dễ dàng tạo ra những giao diện thân thiện và lập trình trên chúng một cách thuận lợi, đa năng.

(5). Cửa sổ Properties – nơi để thiết lập các thuộc tính cho những đối tượng, những điều khiển trong quá trình làm việc trên VB.

b. Lưu trữ

Làm việc trên VB là làm việc trên các dự án (Project). Tại một thời điểm có thể chỉ làm việc với một dự án nào đó, cũng có khi làm việc trên một tập hợp các dự án (Project Group). Tuy nhiên khuôn khổ giáo trình này chỉ nói về làm việc trên một dự án đơn lẻ.

Dự án là công cụ quản lý tất cả những gì cần phải tạo ra cho một dự án phần mềm viết trên VB. Hình dưới mô tả các nội dung có thể được quản lý bởi một dự án mang tên **quan_ly_canbo**.



Trong dự án trên có các thành phần:

Forms – để tạo ra các giao diện người sử dụng phần mềm như là các hộp thoại, biểu nhập dữ liệu, cửa sổ giao diện. Có 4 form được tạo ra trong dự án trên.

Modules – là nơi chứa những thư viện khai báo phục vụ việc phát triển phần mềm. Trong mỗi Module có thể chứa các chương trình con, các khai báo biến, hằng, môi trường làm việc mà các thành phần thư viện này có thể dùng riêng hoặc chia sẻ dùng chung trong toàn bộ dự án. Có 2 module được tạo ra trong dự án trên là Module1 và Module2.

Class Modules – nơi tạo ra các lớp đối tượng do người lập trình tự định nghĩa phục vụ các nhu cầu phát triển riêng. Dự án trên có một tệp lớp là Class1.

User Controls – nơi cho phép người lập trình tự định nghĩa ra các điều khiển phục vụ mục đích công việc riêng để phát triển trong dự án. Dự án trên có một đối tượng điều khiển tự định nghĩa là UserCo.

Designers – nơi tạo ra các môi trường dữ liệu (data environment) và các báo biểu (Data report) phục vụ nhu cầu xử lý, truy xuất và in ấn dữ liệu trong dự án.

Không chỉ dừng lại ở đây, ứng với mỗi dự án trên VB có thể cần tạo ra những đối tượng riêng. Và chúng có thể được quản lý trên cửa sổ Project Explorer.

Để ghi lại một dự án, nhấn thực đơn **File | Save** hoặc nút **Save** trên thanh công cụ hoặc nhấn tổ hợp phím nóng **Ctrl + S**. VB sẽ lần lượt yêu cầu nhập vào tên tệp tin của các đối tượng đã tạo được trên dự án (việc đặt tên này chỉ xuất hiện ở lần ghi đầu

tiên). Tệp tin chính của dự án có phần mở rộng là **.vbp** và biểu tượng như sau:

3. Lập trình VB căn bản

Các thành phần liên quan đến lập trình căn bản trên VB.

- Các kiểu dữ liệu trong VB.
- Sử dụng biến và hằng.
- Các cấu trúc lập trình căn bản.
- Kỹ thuật chương trình con.
- Cách thức soạn thảo chương trình.
- Kỹ thuật bắt lỗi và xử lý lỗi trên VB.

3.1. Kiểu dữ liệu - biến và hằng

a. Kiểu dữ liệu

Cũng như các ngôn ngữ lập trình khác, VB đều hỗ trợ các kiểu dữ liệu cơ bản. Dưới đây giới thiệu chi tiết về từng kiểu.

Boolean

Kiểu lô gíc, tương tự kiểu Boolean trên Pascal. Kiểu này chiếm 2 byte bộ nhớ; chỉ nhận một trong 2 giá trị là: Yes – No hoặc True – False hoặc đôi khi thể hiện dưới dạng số 0 tương đương với False, True tương ứng với bất kỳ số nào khác 0. Khi lập trình CSDL, kiểu Boolean tương ứng với kiểu Yes/No trong bảng dữ liệu.

Byte

Kiểu số nguyên dương trong phạm vi từ 0..255. Kiểu này chiếm 1 byte bộ nhớ.

Integer

Kiểu nguyên, có giá trị trong khoảng -32768...32767. Kiểu này chiếm 2 bytes bộ nhớ.

Long

Kiểu số nguyên dài, có giá trị trong khoảng 2,147,483,648 .. 2,147,483,647. Kiểu này chiếm 4 bytes bộ nhớ.

Single

Kiểu số thực, có giá trị trong khoảng 1.401298E-45 to 3.402823E38. Chiếm 4 bytes bộ nhớ.

Double

Kiểu số thực có độ lớn hơn kiểu Single, có giá trị trong khoảng 4.94065645841247E-324 to 1.79769313486232E308. Chiếm 8 bytes bộ nhớ.

Currency

Kiểu tiền tệ. Bản chất là kiểu số, độ lớn 8 bytes, có giá trị trong khoảng -922,337,203,685,477.5808 to 922,337,203,685,477.5807. Đặc biệt, kiểu này luôn có ký hiệu tiền tệ đi kèm.

String

Kiểu xâu ký tự. Kiểu này tương ứng với kiểu String trong Pascal, tương ứng với kiểu Text trong VB. Độ lớn tối đa 255 bytes tương đương với khả năng xử lý xâu dài 255 ký tự.

Variant

Variant là kiểu dữ liệu không tường minh. Biến kiểu này có thể nhận bất kỳ một giá trị nào có thể. Ví dụ:

Dim a As Variant

a = 123

a = “Bùi Văn Tú”

Hoàn toàn không có lỗi.

Người ta thường khai báo biến kiểu Variant trong những trường hợp phải xử lý biến đó mềm dẻo. Khi thì biến nhận giá trị kiểu này, khi thì nhận giá trị và xử lý theo kiểu dữ liệu khác.

Object

Object là một loại biến kiểu Variant, chiếm dung lượng nhớ 4 bytes, dùng để tham chiếu tới một loại đối tượng (Object) nào đó trong khi lập trình. Tất nhiên muốn khai báo biến Object kiểu nào, phải chắc chắn đối tượng đó đã được đăng ký vào thư viện tham chiếu VB bởi tính năng Project | Reference.

b. Biến

b.1. Biến – khai báo biến

Biến (Variable) là thành phần của một ngôn ngữ lập trình, giúp xử lý dữ liệu một cách linh hoạt và mềm dẻo.

Thông thường trong các ngôn ngữ lập trình, mỗi biến khi tồn tại phải được

định kiểu, tức là phải nhận một kiểu dữ liệu xác định. Tuy nhiên trong VB thì không, mỗi biến có thể định kiểu (được khai báo trước khi sử dụng) hoặc không định kiểu (không khai báo vẫn sử dụng được). Trong trường hợp này biến đó sẽ tự nhận kiểu giá trị Variant.

Biến có thể được khai báo bất kỳ ở đâu trong phần viết lệnh của VB. Tất nhiên, biến có hiệu lực như khai báo chỉ bắt đầu từ sau lời khai báo và đảm bảo phạm vi hoạt động như đã qui định. Vì biến trong VB hoạt động rất mềm dẻo, nên có nhiều cách khai báo biến như:

Ví dụ 1: Khai báo biến i kiểu Integer

Dim i As Interger

Ví dụ 2: Khai báo biến i kiểu Integer, st kiểu String độ dài 15 ký tự

Dim i As Integer, st As String*15

b.2. Phạm vi biến

Như chúng ta đã biết, mỗi biến sau khi được khai báo nó sẽ nhận một kiểu dữ liệu và có một phạm vi hoạt động, tức là lời khai báo biến chỉ có tác dụng trong những vùng đã được chỉ định; ngoài vùng chỉ định đó biến sẽ không có tác dụng, nếu có tác dụng sẽ theo nghĩa khác (biến cục bộ kiểu Variant chẳng hạn).

Biến cục bộ:

Biến cục bộ được khai báo sau từ khoá Dim, nó chỉ có tác dụng trong một chương trình con, cục bộ trong một form hoặc một module nào đó. Dưới đây sẽ chỉ ra 3 trường hợp biến cục bộ này:

- Trong một chương trình con, nếu nó được khai báo trong chương trình con đó;
- Trong cả một Form, nếu nó được khai báo trong phần Decralations của Form đó;
- Trong cả một Reports, nếu nó được khai báo trong phần Decralations của Report đó;
- Trong cả một Modules, nếu nó được khai báo trong phần Decralations của Modules đó;

* Biến chỉ có tác dụng sau lệnh khai báo Dim

Biến toàn cục:

Biến toàn cục được khai báo sau cụm từ khoá Public, nó có tác dụng trong toàn bộ chương trình (ở bất kỳ chỗ nào có thể viết lệnh). Loại biến này luôn phải được khai báo tại vùng Decralations của một Module nào đó.

Ví dụ:

Public Hoten(45) As String * 45

Trên một dự án VB không được phép khai báo trùng tên biến toàn cục. Tuy nhiên tên biến cục bộ vẫn có thể trùng tên biến toàn cục, trong trường hợp đó VB sẽ ưu tiên sử dụng biến cục bộ trong phạm vi của nó.

c. Hằng

c.1. Khai báo hằng

Hằng (Constan) là đại lượng có giá trị xác định và không bị thay đổi trong bất kỳ hoàn cảnh nào. Tương ứng với từng kiểu dữ liệu, sẽ có những hằng tương ứng.

Khai báo hằng số bởi từ khoá Const. Sau đây là các ví dụ về khai báo các loại hằng:

Ví dụ 1: Hằng a =5 (hằng số)

Const a = 5

Ví dụ 2: Hằng ngày = 24/12/2004 kiểu Date (bao bởi cặp dấu thăng #..#)

Const ngay = #24/12/2004#

Ví dụ 3: Hằng xâu ký tự (bao bởi cặp dấu nháy kép “..”)

Const phongban = "Tài vụ"

Ví dụ 4: Hằng kiểu Lôgíc xác định bởi True hoặc False

Const ok = True

c.2. Phạm vi hằng

Tương tự như biến, hằng cũng có những phạm vi hoạt động của nó. Hằng được khai báo trong thủ tục nào, hoặc cục bộ trong form, report hoặc module nào sẽ chỉ có tác dụng trong phạm vi đó.

Muốn hằng có phạm vi toàn cục, phải được khai báo sau từ khoá Public Const, tại vùng Decralations của một module nào đó như sau:

Public Const a = 12

3.2. Các cấu trúc lệnh VB

Các cấu trúc lệnh là thành phần cơ bản của mỗi ngôn ngữ lập trình. Thông thường các ngôn ngữ lập trình đều có các cấu trúc lệnh như nhau: lệnh xử lý điều kiện, lệnh lặp biết trước số vòng lặp, lệnh lặp không biết trước số vòng lặp,.. Tuy nhiên cách thể hiện (cú pháp) mỗi cấu trúc lệnh có thể khác nhau tuỳ thuộc vào mỗi ngôn ngữ lập trình. Hơn nữa, mỗi ngôn ngữ cũng có thể có một số điểm khác biệt, đặc trưng trong mỗi cấu trúc lệnh.

Cũng giống như nhiều ngôn ngữ lập trình hiện đại khác, các cấu trúc lệnh trong VB đều tuân thủ các nguyên tắc:

- Có cấu trúc: mỗi cấu trúc lệnh đều có từ khoá bắt đầu và một từ khóa báo hiệu kết thúc;
- Thực hiện tuần tự (loại trừ trường hợp đặc biệt thủ tục Goto <Label>);
- Có khả năng lồng nhau;

a. Cấu trúc IF… END IF

Cấu trúc này thường gọi là lệnh lựa chọn. Tức là nếu một điều kiện nào đó xảy ra sẽ là gì, hoặc trái lại có thể làm gì. Trong VB cú pháp lệnh này như sau:

```
If <điều kiện> Then
        <thủ tục 1>
[ Else
        <thủ tục 2> ]
End If
```

b. Cấu trúc SELECT CASE .. END SELECT

Đây là một loại của cấu trúc lựa chọn. Thông thường hoàn toàn có thể sử dụng If .. End If để thực hiện các xử lý liên quan đến kiểu cấu trúc này, nhưng trong những trường hợp đặc biệt, cấu trúc Select Case .. End Select thể hiện được sự tiện dụng vượt trội. Trong VB cú pháp lệnh này như sau:

```
Select Case <biểu thức>
    Case <giá trị 1>
            <thủ tục 1>
    Case <giá trị 2>
            <thủ tục 2>
            ……
```

Case <giá trị n>

<thủ tục n>

[Case Else

<thủ tục n+1>]

End Select

Trong đó: <Biểu thức> luôn trả về giá trị kiểu vô hướng đếm được như: số nguyên, xâu ký tự, kiểu lô gíc,..

c. Cấu trúc FOR ... NEXT

For… Next là một cấu trúc lặp biết trước số lần lặp trong VB, tuy nhiên trong những tình huống đặc biệt, vẫn có thể sử dụng cấu trúc này như cấu trúc không biết trước được số lần lặp.

Cú pháp cấu trúc For…Next như sau:

For <biến chạy> = <giá trị 1> To <giá trị 2> [Step <n>]

<thủ tục>

[Exit For]

Next

d. Cấu trúc WHILE ... WEND

While … Wend là một cấu trúc lặp không biết trước số lần lặp trong VB. Cú pháp cấu trúc While…Wend như sau (Wend - viết tắt của cụm từ While End):

While <điều kiện>
<thủ tục>

Wend

3.3. Các hàm xử lý chuỗi trong Vb6

Space (Num as Long)

Trả về chuỗi chỉ toàn khoảng trống với số khoảng trống được ấn định bởi tham số Num.

String (Num as Long, character)

Trả về một chuõi (theo dạng variant) gồm các ký tự lặp lại. Ký tự lặp lại là ký tự đầu của biểu thức chuỗi được truyền ở tham số thứ hai của hàm (character). Tham số thứ nhất (Num) xác nhận số lần lặp lại.

Trim (String)

Cắt các khoảng trống ở 2 đầu chuỗi

Len ( )

Trả về chiều dài của chuỗi bao gồm các khoảng trống và các ký tự.

Mid (string, start as Long, length)

Trích từ tham số 1(string) một chuỗi ở vị trí bắt đầu được xác định bởi tham số 2(start), với số ký tự được qui định bởi tham số 3(length). Nếu bỏ qua tham số length thì hàm Mid sẽ trích đến hết chuỗi.

InStr (start, string1, string2, compare)

Trả về vị trí bắt đầu của một chuỗi con cần tìm trong một chuỗi mẹ. tham số 1(start) xác định vị trí bắt đầu tìm, tham sô 2(string1) là chuỗi mẹ, tham số 3(string2) là chuỗi cần tìm, tham số 4(compare) mặc định là so sánh nhạy ký tự. Khi bỏ qua tham số thứ nhất thì vị trí bắt đầu tìm mặc định là 1.

InStrRev (StringCheck as string, StringMatch as string, Start as Long, Compare)

Chức năng như InStr nhưng InStrRev hoạt động ngược lại từ cuối chuỗi và cú pháp khác hơn. Cả hai hàm đều là hàm tìm kiếm nhạy ký tự nên cần chú ý chữ thường và chữ HOA. InStrRev thường kết hợp với Mid để tách một tên File khỏi đường dẫn và tên mở rộng.

Left (String, Length as Long)

Trích từ đầu một chuỗi của tham số 1(String) với số lượng xác định bởi tham số 2 (Length).

Right (String, Length as Long)

Như Left nhưng trích ngược từ cuối chuỗi.

Replace (Expression as string, Find as string, Replace as string, start, count, compare)

Tìm trong tham số thứ 1(Expression) một chuỗi xác định bởi tham số 2(Find) và thay thế bằng một chuỗi được đặt ở tham số 3(Replace). Ba tham số còn lại là tùy chọn. Start qui định vị trí bắt đầu tìm chuỗi cần được thay, nếu bỏ qua mặc định là 1. Count qui định số lần thay thế trong chuỗi, nếu bỏ qua mặc định Replace sẽ tìm và thay thế cho đến hết chuỗi.

StrComp (String1, String2, Compare)

Dùng để so sánh 2 chuỗi .

Giá trị trả về: (String1 < String2) = -1; (String1 = String2) = 0; (String1 > String2) =1.

Like

So sánh 2 chuỗi cho phép sử dụng biệt ngữ (như dùng ký tự đại diện trong Dos) giá trị trả về = True nếu tương hợp.

Chú ý hàm Like mặc định cũng là hàm nhạy ký tự, theo thiết lập Option Compare ở form hoặc module.

Chr(charcode as Long)

Chuyển mã Ascii thành ký tự.

Asc(String as String)

Trả về mã Ascii của ký tự.

ChrW(charcode)

Chuyển mã Áscii thành ký tự (Hỗ trợ Unicode).

AscW(string)

Chuyển ký tự thành mã Ascii (hỗ trợ Unicode).

Join (SourceArray, Delimiter)

Tạo chuỗi mới từ một mảng chuỗi (SourceArray) với các phần tử được phân định bởi tham số Delimit.

Split (Expression as String, Delimiter, Count, Compare)

Tạo mảng chuỗi từ một chuỗi (Expression). Đặt tham số Delimiter để chuyên biệt chỗ ngắt, nếu bỏ qua tham số này mặc định Split sẽ tách tại các khoảng trống của chuỗi. Tham số Count quy định số lần tách. Ba tham số cuối là tùy chọn.

Filter (sourcearray, match [, include [, compare]])

Lọc mảng sourcesrray với giá trị lọc là match; include: Lọc đảo (True hoặc False); compare: chỉ rõ kiểu dữ liệu để so sánh trong quá trình lọc, dùng cho tham số compare.

vbUseCompareOption = –1 : Chế độ tùy chọn,

vbBinaryCompare = 0: So sánh nhị phân.

vbTextCompare = 1: So sánh chuỗi.

vbDatabaseCompare = 2: So sánh dữ liệu.

StrReverse(expression as String)

Đảo chuỗi expression

Chương 4: CHƯƠNG TRÌNH THỰC NGHIỆM

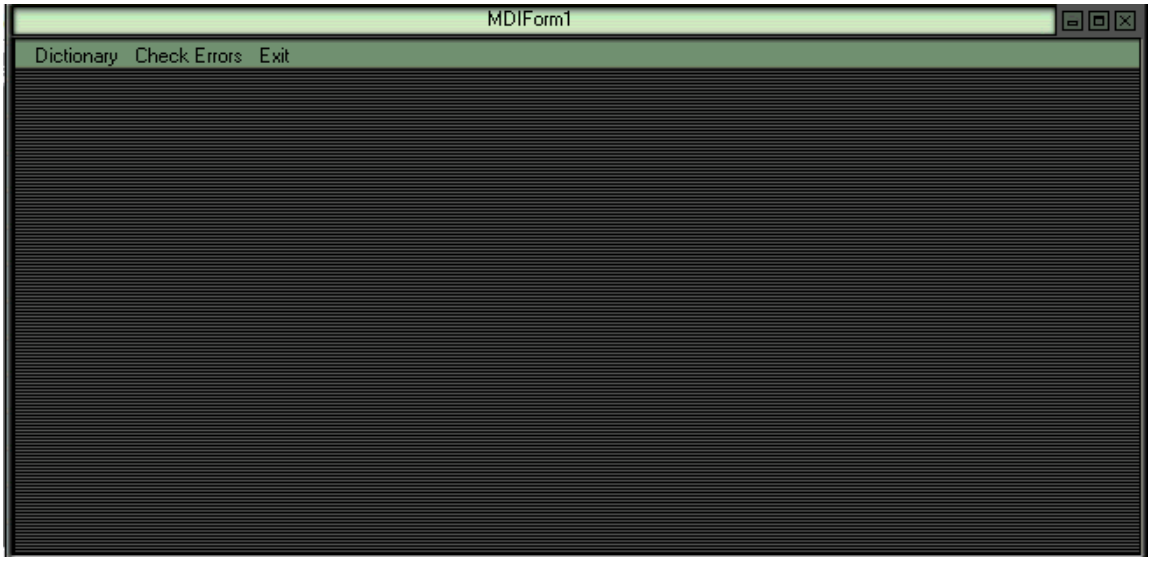
I. Giới thiệu chương trình

- Chữ viết là phương tiện giao tiếp quan trọng của con người và qua đó việc xử dụng sai chữ viết: sai từ, sai câu… dễ dẫn tới hậu quả nghiêm trọng trong việc thể hiện điều muốn diễn đạt. Trong khi, lỗi khi xử dụng từ, câu là không thể tránh khỏi, nhất là đối với những người mới học tiếng nước ngoài.

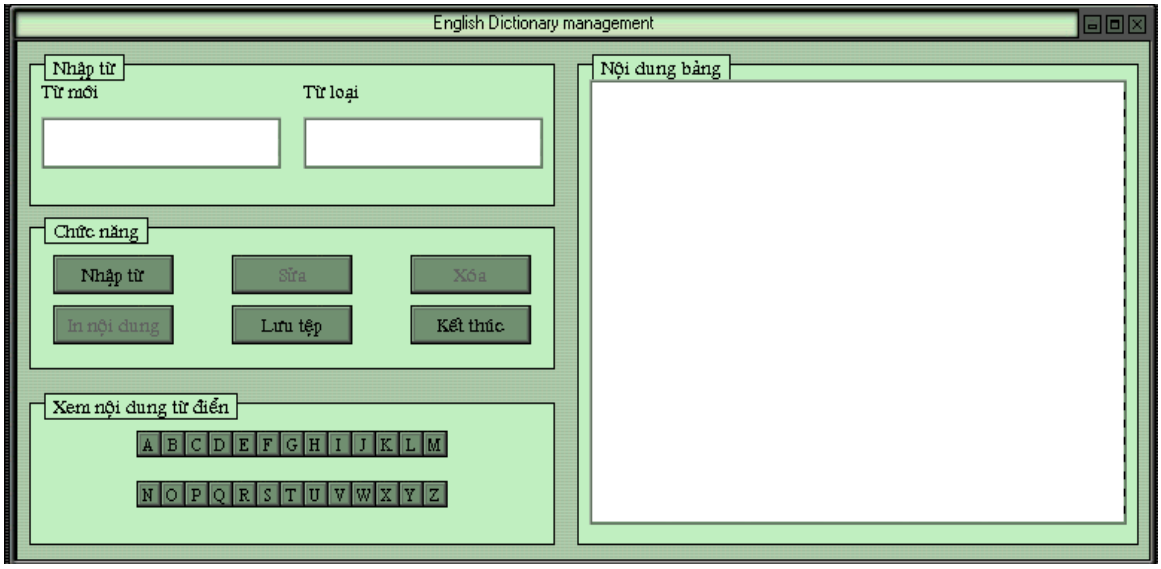
- Chương trình mô phỏng tìm lỗi từ vựng trong việc sử dụng câu tiếng anh là một lĩnh vực trong chương trình xử lý ngôn ngữ tự nhiên. Việc tìm ra lỗi trong sử dụng câu tiếng Anh sẽ đóng góp cho quá trình sửa lỗi từ, câu giúp ích cho người mới học tiếng Anh hay có thể là cơ sở lập trình cho những công việc khác trong lĩnh vực xử lý ngôn ngữ tự nhiên.

- Chương trình được viết bằng ngôn ngữ lập trình Visual Basic 6.0 thuộc bộ Visual Studio 6.0 của Microsoft. Bằng phương pháp khai thác dữ liệu từ điển, chương trình viết ra đã giải quyết được mốt số yêu cầu đặt ra trong việc tìm lỗi khi sử dụng câu tiếng Anh.

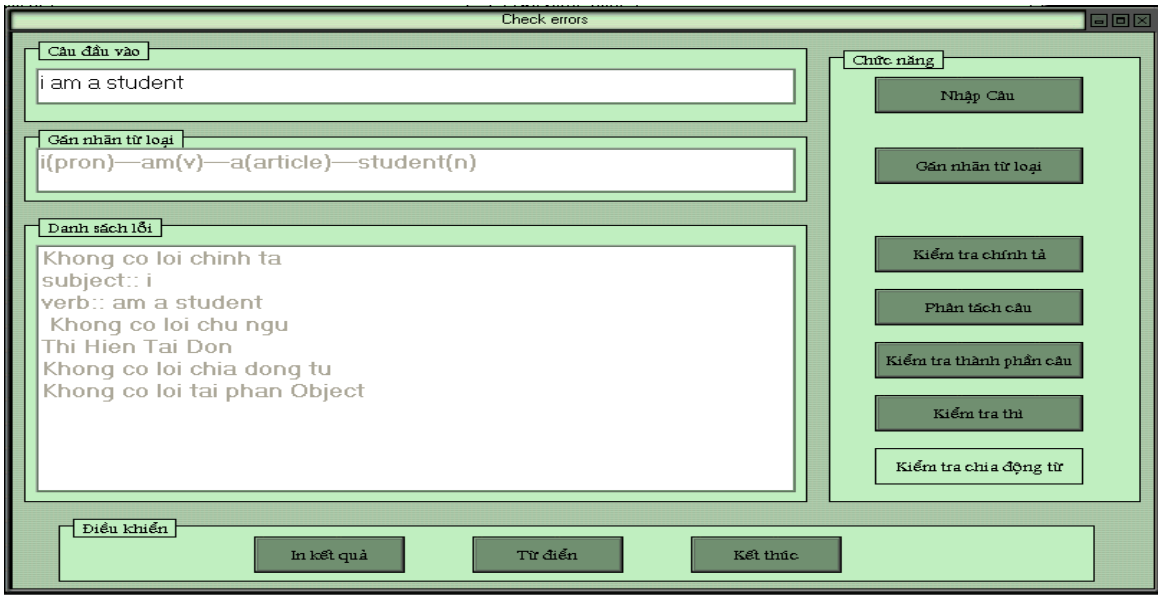
- Chương trình được thiết kế bao gồm 3 Form đóng vai trò khác nhau trong việc giải quyết bài toán.



1. MDI Form



2. *Form Quản lý từ điển*



3. *Form kiểm tra lỗi*

II. Phát biểu bài toán

1. Dữ liệu đầu vào:

Dữ liệu nhập vào của chương trình là một câu đơn tiếng anh theo mẫu.

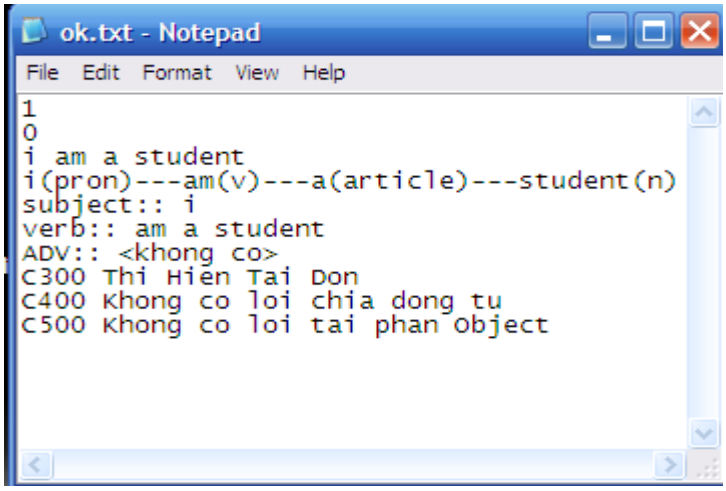
Subject + Verb

Ví dụ: *I am a students*

2. Kết quả của chương trình:

Chương trình sẽ thông báo cho người sử dụng biết các lỗi từ vựng của câu (nếu có) thông qua bảng danh sách lỗi và có thể kết xuất kết quả này ra một file text

File kết quả



III. Tư tưởng, chiến lược

1. Tư tưởng

- Xây dựng bộ từ điển tiếng Anh bao gồm các từ và từ loại của chúng.
- Dựa vào vị trí động từ trong câu để phân tách câu.
- Dựa vào động từ để xác định thì của câu.
- Dựa vào cách sử dụng các thì để kiểm tra lỗi có thể xảy ra ở mỗi vị trí trong câu (S, V, O).
- Kiểm tra lỗi trạng ngữ căn cú vào vị trí trạng ngữ trong câu.
- Khó nhất: Kiểm tra Object.

2. Chiến lược.

* Bottom – Up:

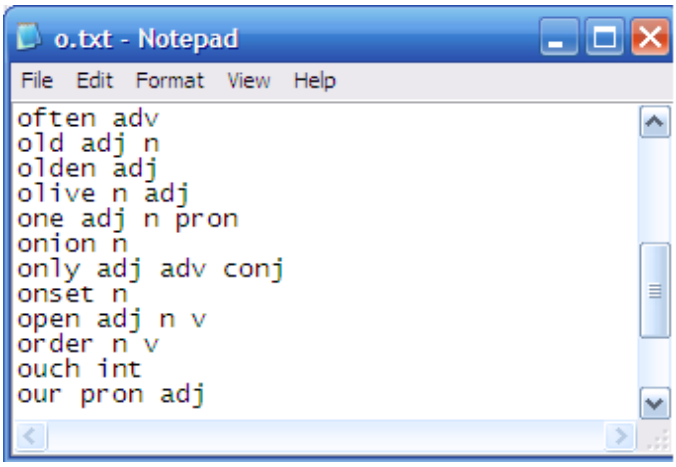
- Xây dựng chương trình từ các module với các chức năng cụ thể, riêng biệt.
- Các module được xây dựng với các hàm thiết thực nhằm thực hiện tư tưởng bài toán.

IV. Bộ dữ liệu từ điển

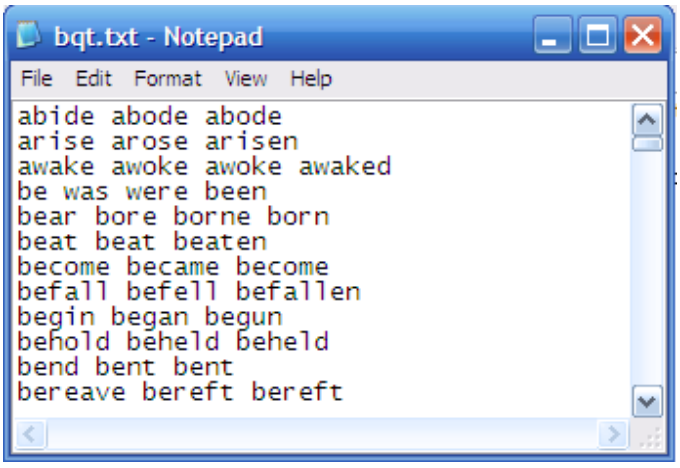
1. Giới thiệu.

Bộ từ điển là các file dữ liệu dang "filename.txt" bao gồm:

+ 26 file chứa các từ bắt đầu bằng các chữ cái trong bảng chữ cái tiếng Anh từ A đến Z và từ loại của chúng.



+ Một file gồm các động từ bất quy tắc và quá khứ phân từ.



2. Ý nghĩa.

Theo tư tưởng đã xác định, bộ từ điển là dữ liệu quan trọng nhất trong bài toán em xây dựng.

Các thuật toán giải quyết bài toán phải phù hợp và tuân theo sự chính xác của từ điển.

3. Ưu, nhược điểm:

Ưu điểm:

- Nhẹ, thao tác dễ dàng trong ngôn ngữ lập trình VB.

- Không đòi hỏi hệ quản trị cơ sở dữ liệu cao cấp -> dễ dàng hơn cho người lập trình và tiếp nhận phần mềm.

Nhược điểm:

- Dễ sai sót, đòi hỏi sự tỉ mỉ và chính xác cao.

- Tính bảo mật yếu.

V. Chương trình.

1. Hệ thống Module thuật toán.

1.1 Chuẩn hóa

* Chuẩn hóa:

Public Static Function chuanhoa(st As String) As String

+ Input: Một chuỗi st bất kỳ.

+ Output: Chuỗi in thường và 2 đầu không có khoảng trắng.

+ Thuật toán:

"st = Trim$(st)

st = LCase(st)

chuanhoa = st"

1.2 Thao tác file.

Bao gồm các hàm xử lý file.

a, Check file

Public Static Function checkfile(checkword As String) As Boolean

+ Input: Một từ tiếng Anh *checkword*

+ Output: Kiểm tra từ có trong từ điển hay không.

Không có trong từ điển checkfile = False

+ Thuật toán:

- Lấy chữ cái đầu của checkword gán = st
- Tìm trong file st.txt xem có tù checword hay không. Nếu có checkfile=true, ngược lại checkfile=false.

b, Check and post

Public Static Function checkandpost(checkword As String) As String

+ In put: Một từ *checkword.*

+ Output: Từ và từ loại của từ.

+ Thuật toán:

- Kiểm tra checkword có trong từ điển hay không. Nếu có, lấy nhãn tù loại và gán cho nó.

1.3 Từ loại:

a, Tính từ

Public Static Function Tinhtu(ByVal word As String) As Boolean

+ Input: Từ *word*

+ Output: Word là tính từ: Tinhtu(checkword) = true

Word không là tính từ: Tinhtu(checkword) = False

+ Thuật toán:

- Tìm đến dòng chứa checkword trong từ điển.
- Lấy từng từ trong dòng để kiểm tra.
- Nếu có từ = "adj" -> tinhtu(checkword) = true

b, Động từ

Public Static Function dongtu(ByVal word As String) As Boolean

Tương tự với tính từ, chỉ thay "adj" = "v".

c, Danh từ

Public Static Function danhtu(ByVal word As String) As Boolean

Tương tự với tính từ, chỉ thay "adj" = "n".

d, Đại từ

Public Static Function daitu(ByVal word As String) As Boolean

Tương tự với tính từ, chỉ thay "adj" = "pron".

e, Mạo từ

Public Static Function maotu(ByVal word As String) As Boolean

Tương tự với tính từ, chỉ thay "adj" = "article".

f, Liên từ

Public Static Function lientu(ByVal word As String) As Boolean

Tương tự với tính từ, chỉ thay "adj" = "conj".

g, Tính trạng từ

Public Static Function tinhtrangtu(ByVal word As String) As Boolean

+ Input: Từ "word" .

+ Output: Word là tính trạng từ: Tinhtrangtu(word) = true

Word không là tính trạng từ: Tinhtrangtu(word) = False

+ thuật toán:

- Lấy 2 ký tự cuối của từ, so sánh với "ly".
- Nếu true, kiểm tra *"st" = "word" – "ly"* nếu: Tinhtu(st) = true -> tinhtrangtu(word) = true; ngược lại tinhtrangtu(word) = false.
- Nếu false, tinhtrangtu(word) = False

g, Danh động từ

Public Static Function danhdongtu(ByVal word As String) As Boolean

+ Input: Từ "word" .

+ Output: Word là tính trạng từ: danhdongtu(word) = true

Word không là tính trạng từ: danhdongtu(word) = False

+ thuật toán:

- Lấy 3 ký tự cuối của từ, so sánh với "ing".
- Nếu true, kiểm tra *"st" = "word" – "ing"* nếu: dongtu(st) = true -> danhdongtu(word) = true; ngược lại danhdongtu(word) = false.
- Nếu false, danhdongtu(word) = False

h. Động từ nguyên thể

Public Static Function dongtunguyenthe(ByVal word As String) As String

+ Input: Từ "word" .

+ Output: Trả về dạng nguyên thể của "word"

+ Thuật toán:

- Chỉ nhận từ có kết thúc bằng đuôi "ed" hoặc "ing".
- Kiểm tra st = "word" – ("ed" or "ing") là động từ -> st là động từ nguyên thể.

i, Từ quá khứ

Public Static Function tuquakhu(ByVal word As String) As Boolean

+ Input: Từ "word" .

+ Output: Kiểm tra xem "word" có là từ quá khứ hay không.

+ Thuật toán:

- st = dongtunguyenthe(word)
- tuquakhu(word) = checkfile(st)

j, Bất quy tắc

Public Static Function batquytac(ByVal word As String) As Boolean

--> Nếu "word" có trong bảng bất quy tắc trong bộ từ điển ->batquytac(word) = true, ngược lại = false.

k, Động từ nguyên thể

Public Static Function verbnguyenthe(ByVal word As String) As Boolean

--> Kiểm tra nguyên thể của động từ có đuôi "s' hoặc "es".

l, Nguyên thể của động từ bất quy tắc

Public Static Function VerbnguyentheBQT(ByVal word As String) As Boolean

--> Kiểm tra nguyên thể của động từ bất quy tắc.

m, Quá khứ động từ thường

Public Static Function verbquakhu(ByVal word As String) As Boolean

--> Kiểm tra thì quá khứ của động từ thường

n, Quá khứ động từ bất quy tắc

Public Static Function verbquakhuBQT(ByVal word As String) As Boolean

--> Kiểm tra thì quá khứ của động từ bất quy tắc.

o, Động từ phân từ 2 bất quy tắc

Public Static Function verbp2BQT(ByVal word As String) As Boolean

--> Kiểm tra thì quá khứ của động từ phân từ 2 bất quy tắc.

1.4 Xử lý câu

a, Gán nhãn từ loại

Public Static Function gannhan(ByVal sentence As String) As String

--> Dựa vào từ điển lấy ra từ và từ loại của từ.

b, Kiểm tra động từ

Public Static Function VerbKT(ByVal st As String) As Boolean

--> "st" là động từ thường, động từ bất quy tắc, hay phân từ 2 thì *VerbKT(st)*= true

c, Tìm chốt

Public Static Function timchot(ByVal cau, ByVal tag) As Integer

--> Lấy vị trí của động từ trong câu.

d, Chia câu làm 2 phần.

Public Static Function chiacau2(ByVal cau, ByVal tag) As Boolean

e, Kiểm tra dấu phẩy

Public Static Function kiemtracom(ByVal cau) As Integer

--> Kiểm tra vị trí dấu phẩy trong câu (nếu có).

f, Kiểm tra trạng ngữ.

Public Static Function trangngudau(ByVal cau, ByVal tag) As Boolean

Public Static Function trangngugiua(ByVal cau, ByVal tag) As Boolean

Public Static Function trangngucuoi(ByVal cau, ByVal tag) As Boolean

--> Định nghĩa vị trí trạng ngữ.

g, Phân tách câu

Public Static Function phantachcau(ByVal cau, ByVal tag) As Boolean

--> Tách câu làm 3 phần: Chủ ngữ, trạng ngữ, động từ + tân ngữ.

h, Kiểm tra lỗi trạng ngữ.

Public Static Function kiemtratrangngu(ByVal tra) As Boolean

Public Static Function KTtrangngugiua(ByVal tra) As Boolean

i, Kiểm tra thì của câu

+ Hiện tại đơn:

Public Static Function kthientaidon(ByVal arr) As Boolean

-- > Định nghĩa theo cấu trúc câu:

S+ V…(Trong đó S là chủ ngữ, V là động từ thường)

* Nếu chủ ngữ là ngôi thứ 3 số ít(He,She, It, hoặc là một danh từ) thì động từ phải thêm “S” hoặc “ES”

+ Thì hiện tại tiếp diễn(The present continuous/progressive tense)

Public Static Function kthientaitiepdien(ByVal arr) As Boolean

-- > Định nghĩa theo cấu trúc câu:

S + am/is/are + V_ing…

+ Thì hiện tại hoàn thành(The Present Prefect Tense)

Public Static Function kthientaihoanthanh(ByVal arr) As Boolean

--> Định nghĩa theo cấu trúc câu:

S + have/has + PP… (PP : Quá khứ phân từ)

* Nếu chủ ngữ là ngôi thứ 3 số ít thì chúng ta dùng "has".

+ Thì hiện tại hoàn thành tiếp diễn(The Present Prefect continuousTense)

Public Static Function kthientaihoanthanhtiepdien(ByVal arr) As Boolean

--> Định nghĩa theo cấu trúc câu:

S + have/has + been + V_ing…

+ Thì quá khứ đơn(The Simple Past Tense)

Public Static Function ktquakhudon(ByVal arr) As Boolean

--> Định nghĩa theo cấu trúc câu:

S + V_ed/V2…

* Nếu là động từ có quy tắc thì thêm "ED" vào sau động từ thường, nếu là động từ bất quy tắc thì sử dụng động từ ở cột 2 trong bảng động từ bất quy tắc.

+ Thì quá khứ tiếp diễn (The Past continuous Tense)

Public Static Function ktquakhutiepdien(ByVal arr) As Boolean

--> Định nghĩa theo cấu trúc câu:

S + was/were + V_ing…

Was: dùng cho ngôi I và ngôi thứ 3 số ít.

+ Thì tương lai đơn(The Simple Future Tense)

Public Static Function kttuonglaidon(ByVal arr) As Boolean

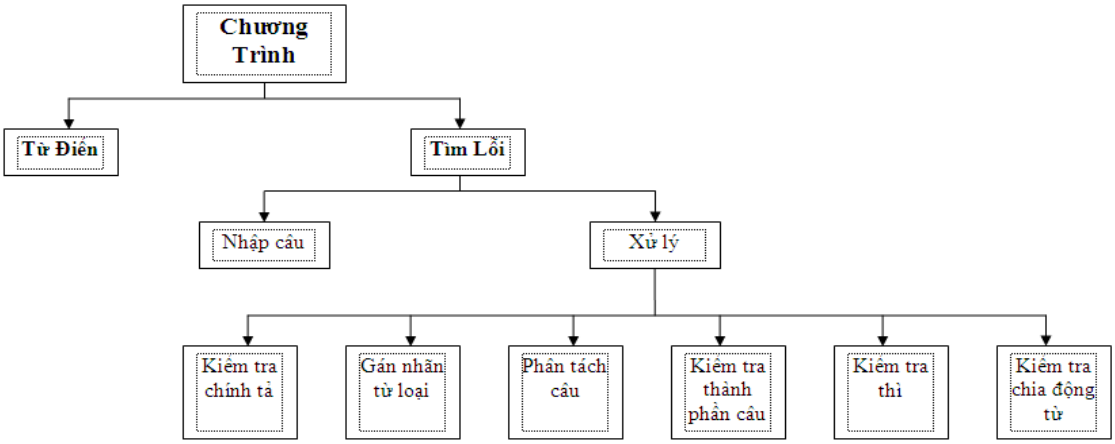
--> Định nghĩa theo cấu trúc câu:

S + will/shall + V …

* Shall được dùng cho ngôi I và We. Trong văn nói và trong tiếng anh ngày nay người ta sử dụng "will" cho tất cả các ngôi.

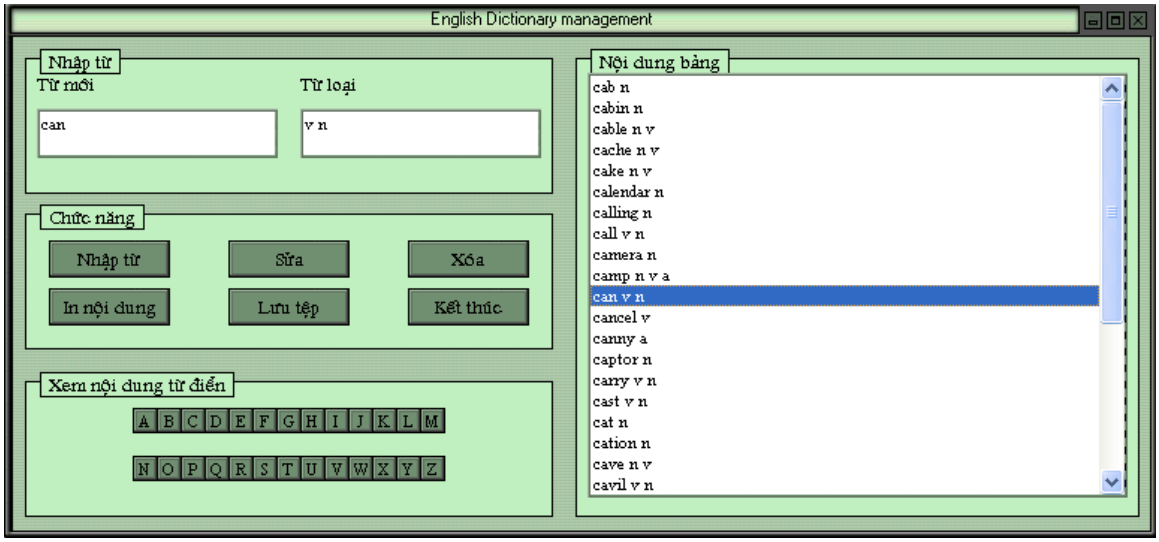
2. Giao diện và chức năng.

a, Mô hình tổng quan



b, Mô tả chức năng

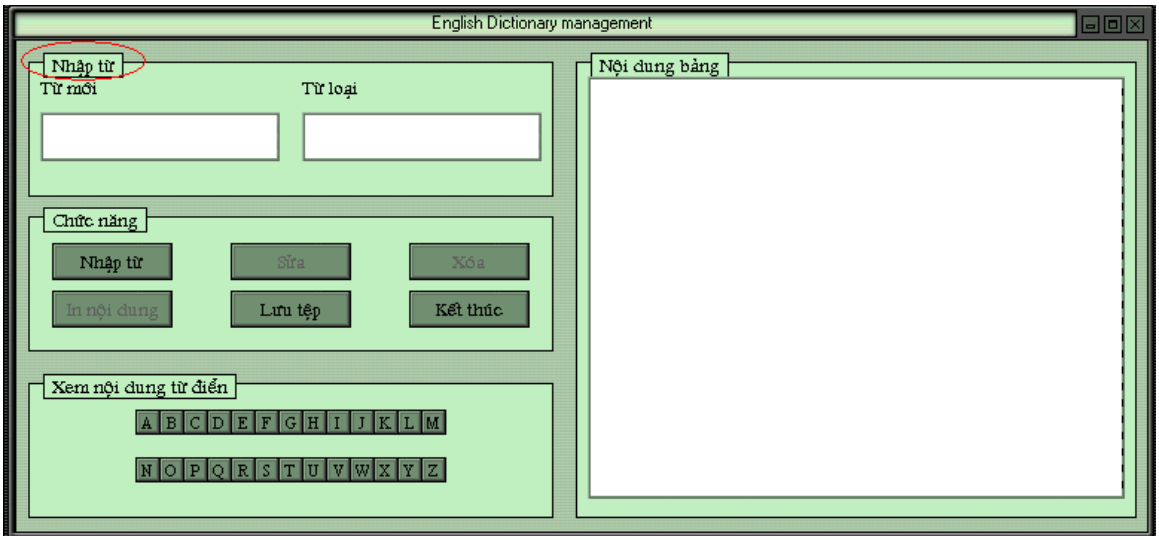
b.1. Từ điển:



Giao diện Form từ điển của chương trình

** Nhập từ*

Cho phép nhập từ mới vào từ điển:



Nhập từ mới vào từ điển bạn phải nhập chính xác từ và từ loại của nó, từ loại được nhập bàng các ký hiệu viết tắt như sau:

"n" : Danh từ

"adj" : Tính từ

"adv" : Trạng từ

"artice": Mạo từ

"pron" : Đại từ

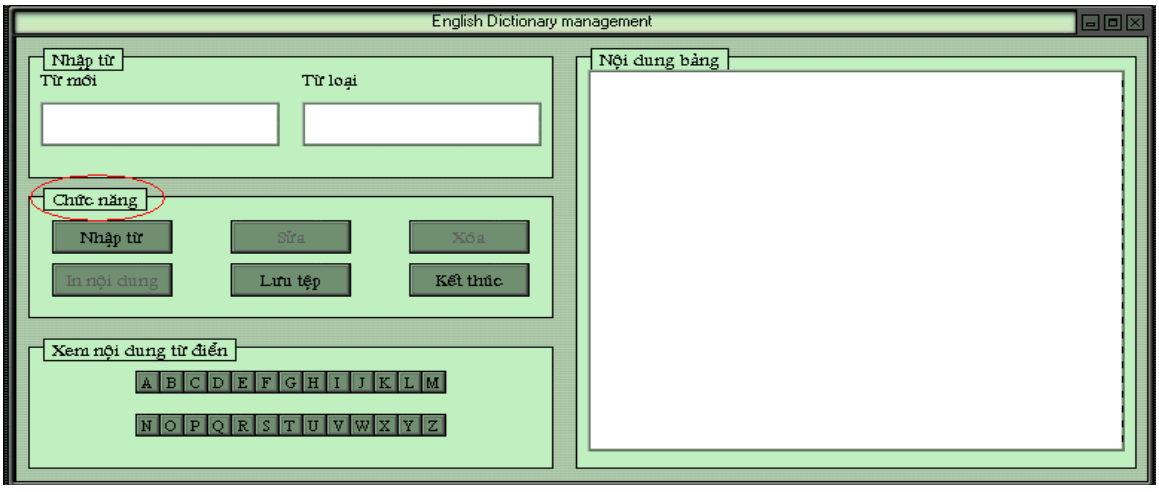
"prep" : Giới từ

"conj" : Liên từ

"v" : Động từ

Một từ có thể thuộc nhiều từ loại, khi đó khi nhập các từ loại cách nhau một dấu cách.

** Chức năng*



Các chức năng:

1. *Nhập từ:*

- Cho phép nhập từ mới và từ loại của nó vào cơ sở dữ liệu từ điển, từ và từ loại sẽ được nhập vào dòng cuối cùng trong file chứa từ điển.

```
"Private Sub Command1_Click()
    Dim temp As String
    Dim temp1 As String
    temp = Text1.Text & " " & Text2.Text
    temp = chuanhoa(temp)
    temp1 = Mid(temp, 1, 1)
    Open App.Path & "\" & temp1 & ".txt" For Append As #1
    Print #1, temp
    Close #1
    List1.Clear
    Open App.Path & "\" & temp1 & ".txt" For Input As #1
    While Not EOF(1)
    Line Input #1, temp
    List1.AddItem temp
    Wend
    Close #1
    Command4.Enabled = True
End Sub"
```

2. *Sửa:*

- Cho phép sửa từ và từ loại trong từ điển

```
"Private Sub Command2_Click()
    Dim st As String
    If (Text1.Text = "") And (Text2.Text = "") Then
    List1.RemoveItem (inn)
    Else
    List1.RemoveItem (inn)
    st = Text1.Text + " " + Text2.Text
```

List1.AddItem st, inn

End If

kk = False

Command4.Enabled = True

End Sub"

3. <u>Xóa</u>

- Xóa từ trong từ điển.

"Private Sub Command3_Click()

List1.RemoveItem (inn)

kk = False

End Sub"

4. <u>In nội dung</u>

- In nội dung từ điển ra file mới.

"Private Sub Command4_Click()

Dim filename As String

Dim i As Integer

conn.ShowSave

filename = conn.filename + ".txt"

Open filename For Output As #1

For i = 0 To List1.ListCount - 1

Print #1, List1.List(i)

Next

Close #1

End Sub"

5. <u>Lưu tệp</u>

- Lưu lại thông tin đã sửa hoặc xóa từ

"Private Sub Command5_Click()

Dim i As Integer

If MsgBox("Are you sure?", vbOKCancel, Warning) = vbOK Then

Open App.Path & namefi For Output As #1

```
                For i = 0 To List1.ListCount - 1
                        Print #1, List1.List(i)
                Next
        Close #1
        End If
        kk = True
End Sub"
```

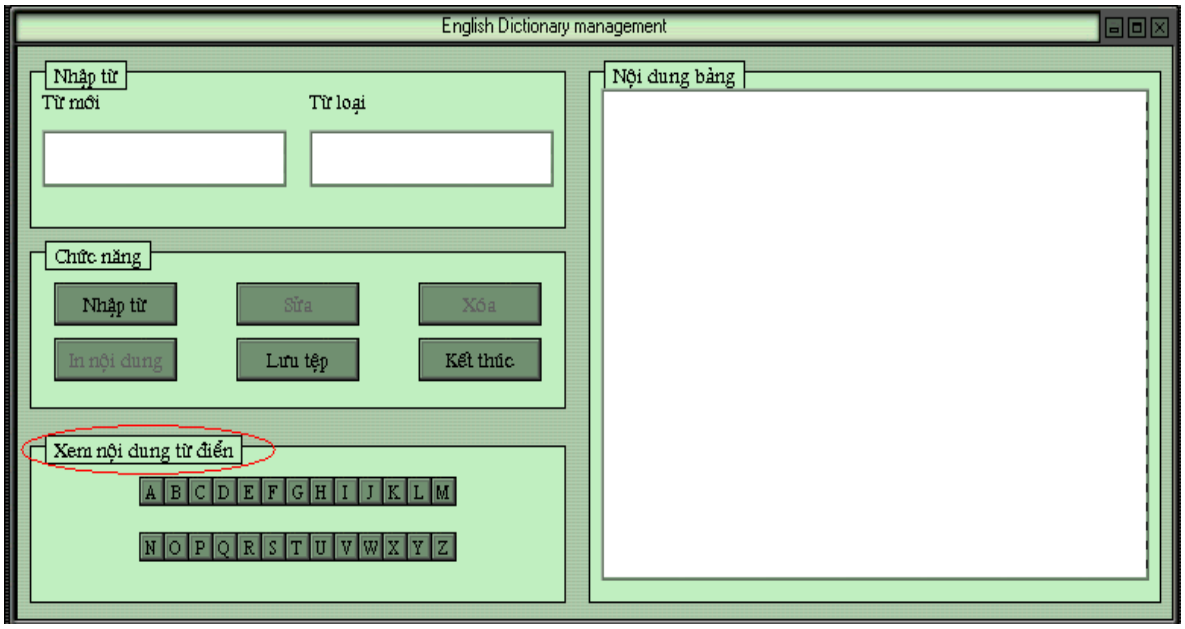
6. <u>*Kết thúc*</u>

- Đóng form, kết thúc làm việc.

```
"Private Sub Command6_Click()
        If kk = True Then
        Unload Me
        Else
                If MsgBox("Chua ghi noi cung can thay doi. Ban co muon ghi
        khong?", vbOKCancel, Warning) = vbOK Then
        Open App.Path & namefi For Output As #1
                For i = 0 To List1.ListCount - 1
                   Print #1, List1.List(i)
                Next
        Close #1
        kk = True
        Unload Me
        Else
        kk = True
        Unload Me
        End If
        End If
End Sub"
```

* *Xem nội dung từ điển:*

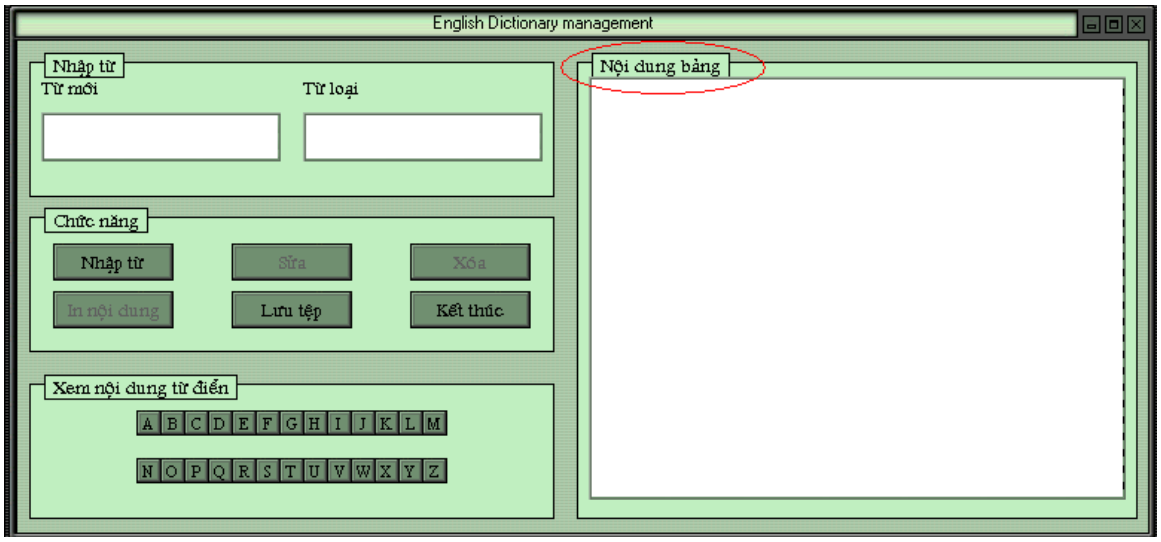
Xem nội dung từ điển theo chữ cái đầu.




```
Private Sub Command7_Click()
    Dim temp As String
    Text1.Text = ""
    List1.Clear
    namefi = "\a.txt"
    Open App.Path & "\a.txt" For Input As #1
    While Not EOF(1)
        Line Input #1, temp
        List1.AddItem temp
        Wend
    Close #1
    Command4.Enabled = True
End Sub
```

* *Nội dung bảng:*

Hiển thị nôi dung từ điển trong suốt quá trình làm việc với từ điển theo chữ cái đầu của từ.



“*Private Sub List1_Click()*

Dim st As String

Dim st1 As String

Dim arr

Dim i As Integer

Dim j As Integer

st = List1.Text

arr = Split(st, " ")

Text1.Text = arr(0)

i = UBound(arr)

j = 1

st1 = ""

While j <= i

st1 = st1 + arr(j) + " "

j = j + 1

Wend

Text2.Text = chuanhoa(st1)

inn = List1.ListIndex

Command2.Enabled = True

Command3.Enabled = True

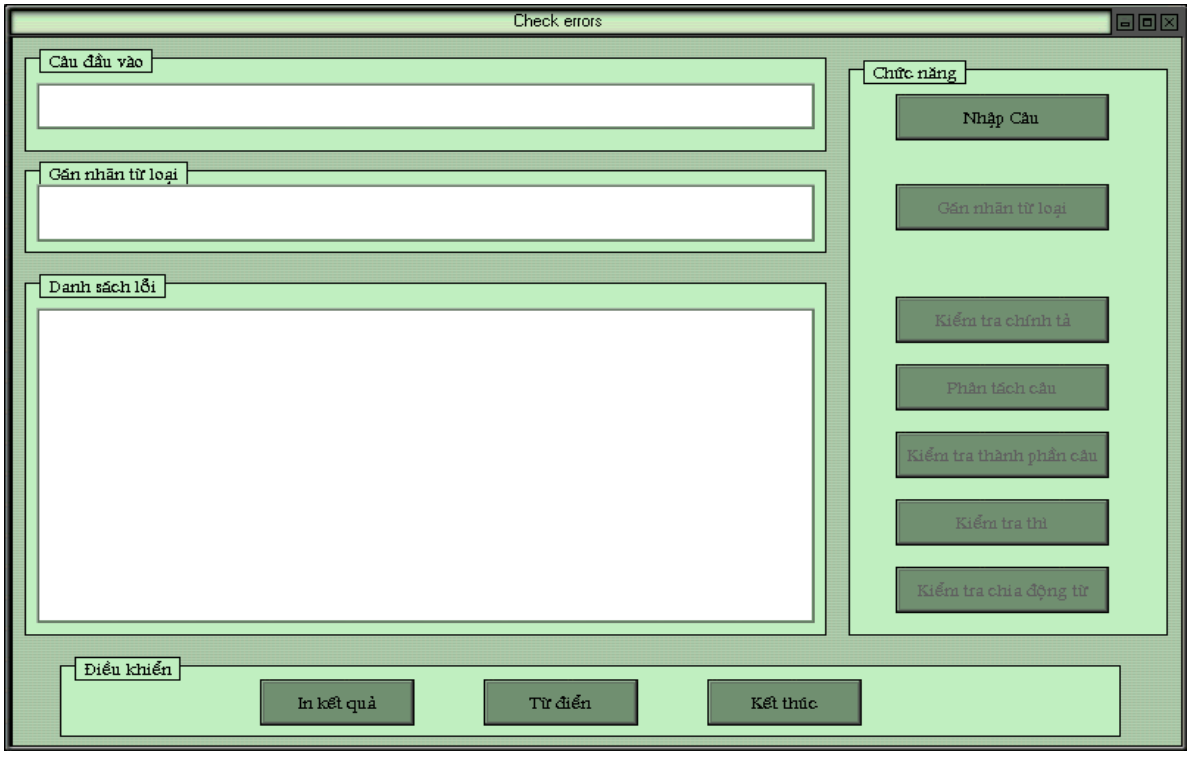
End Sub"

b.2. Tìm Lỗi câu

Input: Một câu tiếng Anh (câu đơn, khẳng định)

Output:

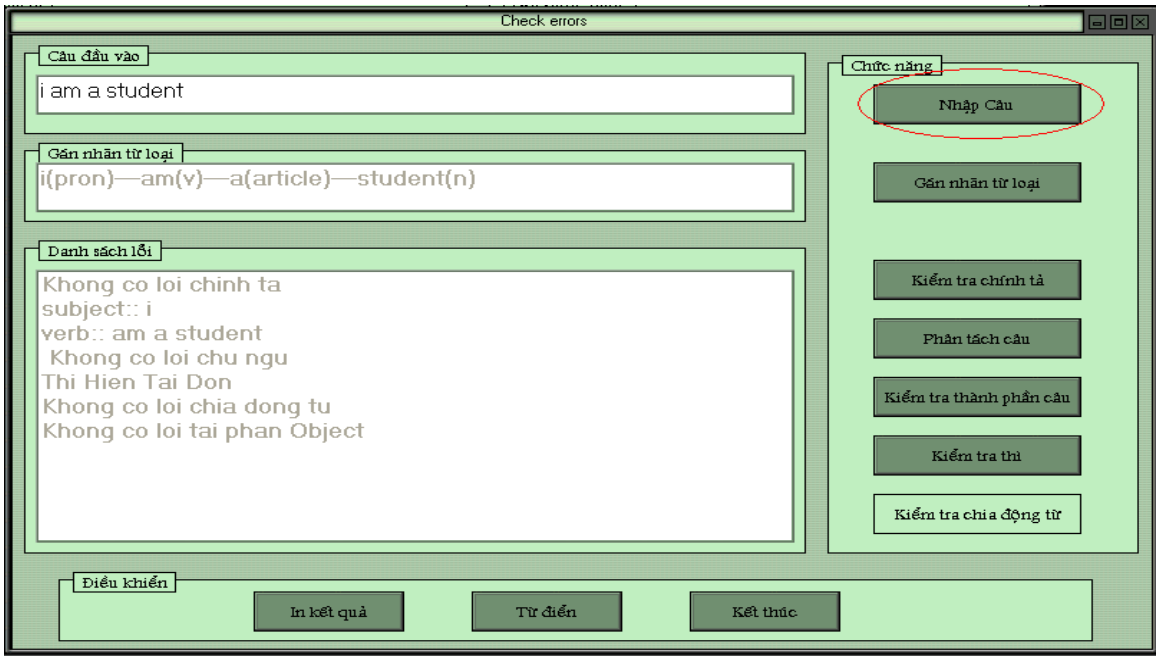
- Gán nhãn từ loại.
- Kiểm tra lỗi sai chính tả.
- Phân tách câu: S + (V + O).
- Kiểm tra thành phần câu: kiểm tra lỗi chủ ngữ.
- Kiểm tra thì.
- Kiểm tra lỗi chia động từ trong câu.



Giao diện Form Tìm Lỗi câu

** Nhập câu:*

Nhận câu đầu vào để xử lý.



"Private Sub Command4_Click()

If Text1.Text <> "" Then

cauday = chuanhoa(Text1.Text)

Command6.Enabled = True

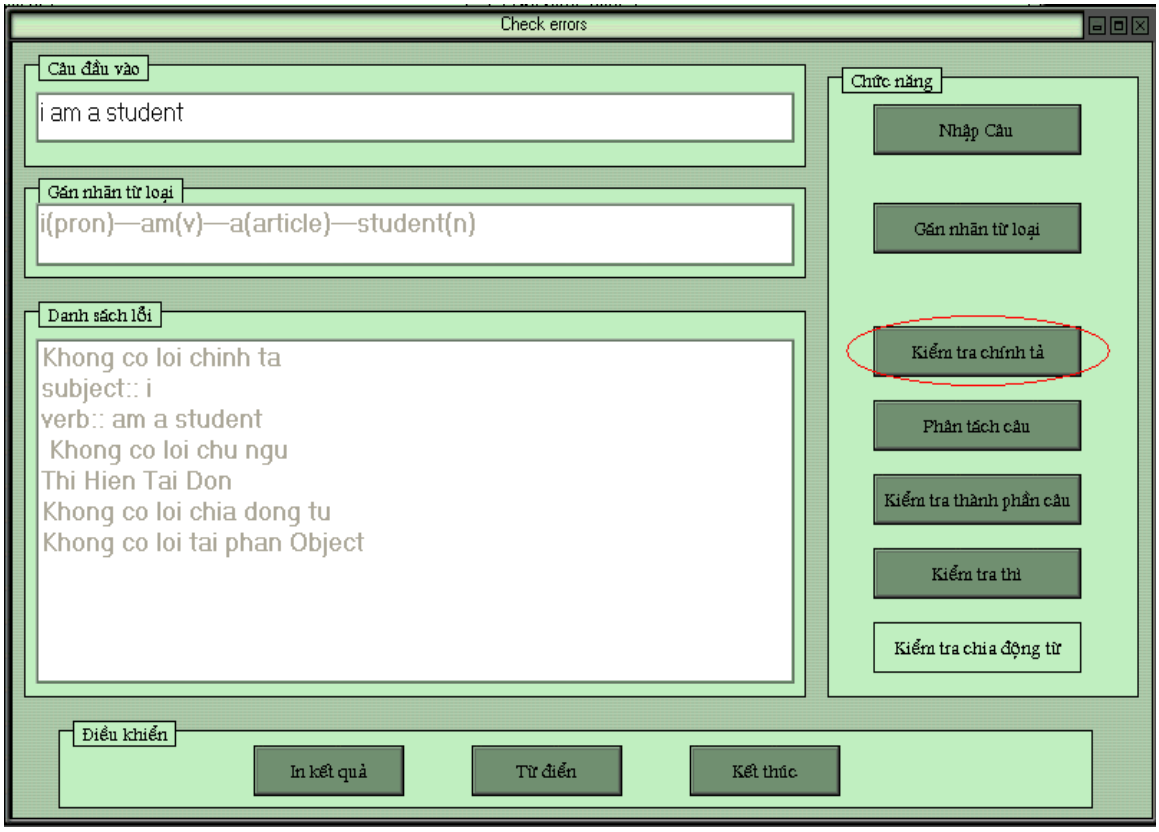
List1.Clear

End If

End Sub"

** Kiểm tra chính tả:*

Kiểm tra xem câu có sai lỗi chính tả hay không. Nếu sai thì thông báo sai lỗi chính tả hoặc từ không có trong dữ liệu từ điển, bạn có thể mở giao diện từ điển để nhập hoặc sửa từ thông qua điêu khiển "Từ điển".



"*Private Sub Command6_Click()*

Dim cau As String

Dim danhsach As String

Dim arr

Dim i As Integer

Dim j As Integer

Dim check As Boolean

Dim st As String

Dim st1 As String

check = False

cau = cauday

arr = Split(cau, " ")

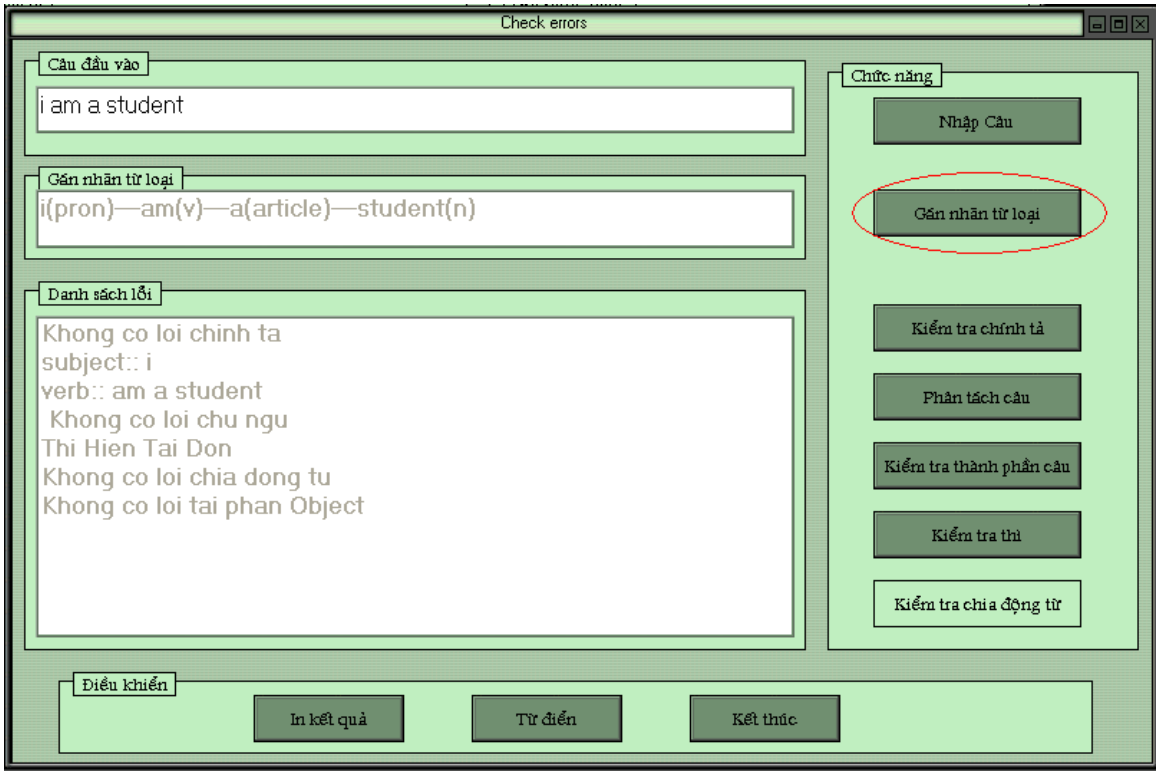
i = UBound(arr)

```
j = 0
danhsach = ""
While j <= i
   st = arr(j)
   check = checkfile(st)
   If check = True Then
      GoTo a:
   Else
      check = tinhtrangtu(st)
      If check = True Then
      GoTo a:
   End If
   check = danhdongtu(st)
   If check = True Then
      GoTo a:
   End If
   check = tuquakhu(st)
   If check = True Then
        GoTo a:
   End If
        check = batquytac(st)
   If check = True Then
        GoTo a:
   End If
        check = danhtunhieu(st)
   If check = True Then
        GoTo a:
   End If
   End If
a:
  If check = True Then
```

```
        j = j + 1
      Else
        danhsach = danhsach + st + " "
        j = j + 1
      End If
    Wend
    If danhsach = "" Then
      st1 = "Khong co loi chinh ta"
    List1.AddItem st1
    Command5.Enabled = True
    Else
      st1 = danhsach + "khong co trong tu dien hoac sai chinh ta"
    List1.AddItem st1
    End If
End Sub"
```

<u>* Gán nhãn từ loại</u>

Từ không sai chính tả trong câu được gán nhãn theo từ điển.



"Private Sub Command5_Click()

Dim st As String

st = cauday

Text2.Text = gannhan(st)

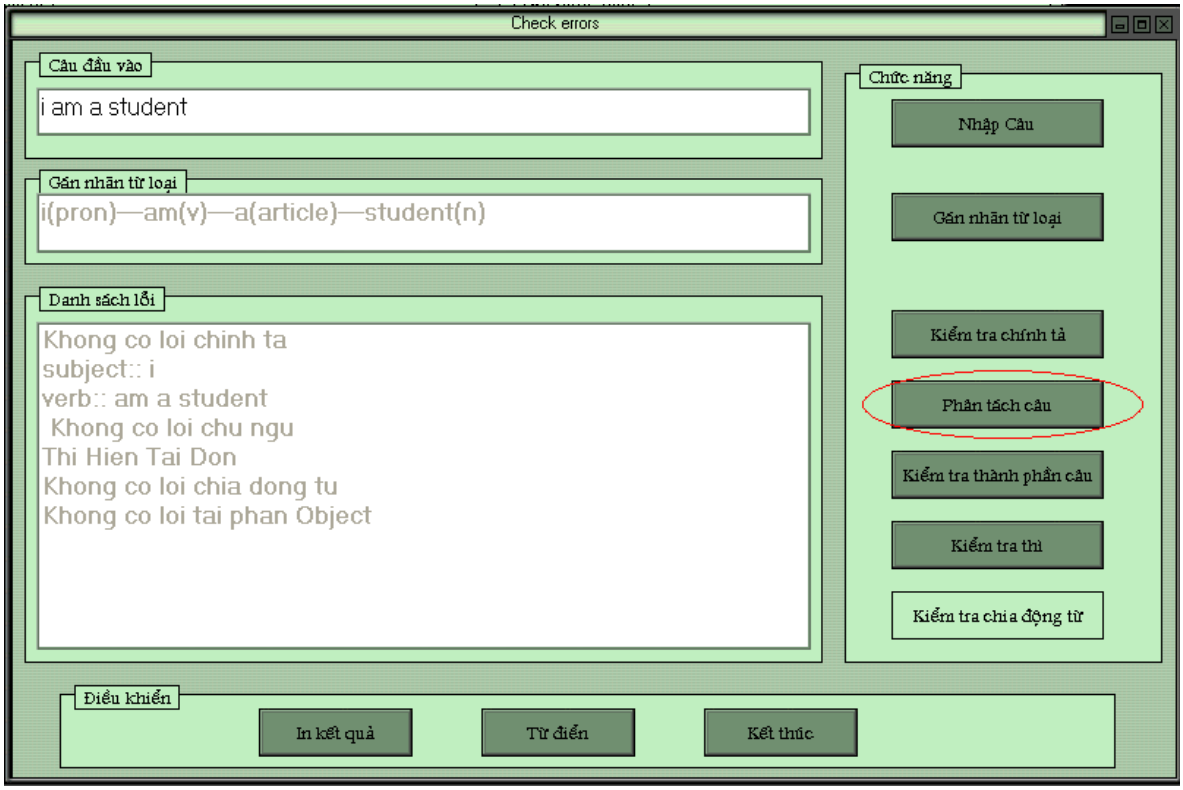
Command7.Enabled = True

End Sub"

** Phân tách câu*

Câu được phân tách thành 2 phần:

S + (V +O)



"Private Sub Command7_Click()

Dim i As Boolean

Dim st As String

Dim j As Integer

Dim k As Integer

i = phantachcau(tu, tagg)

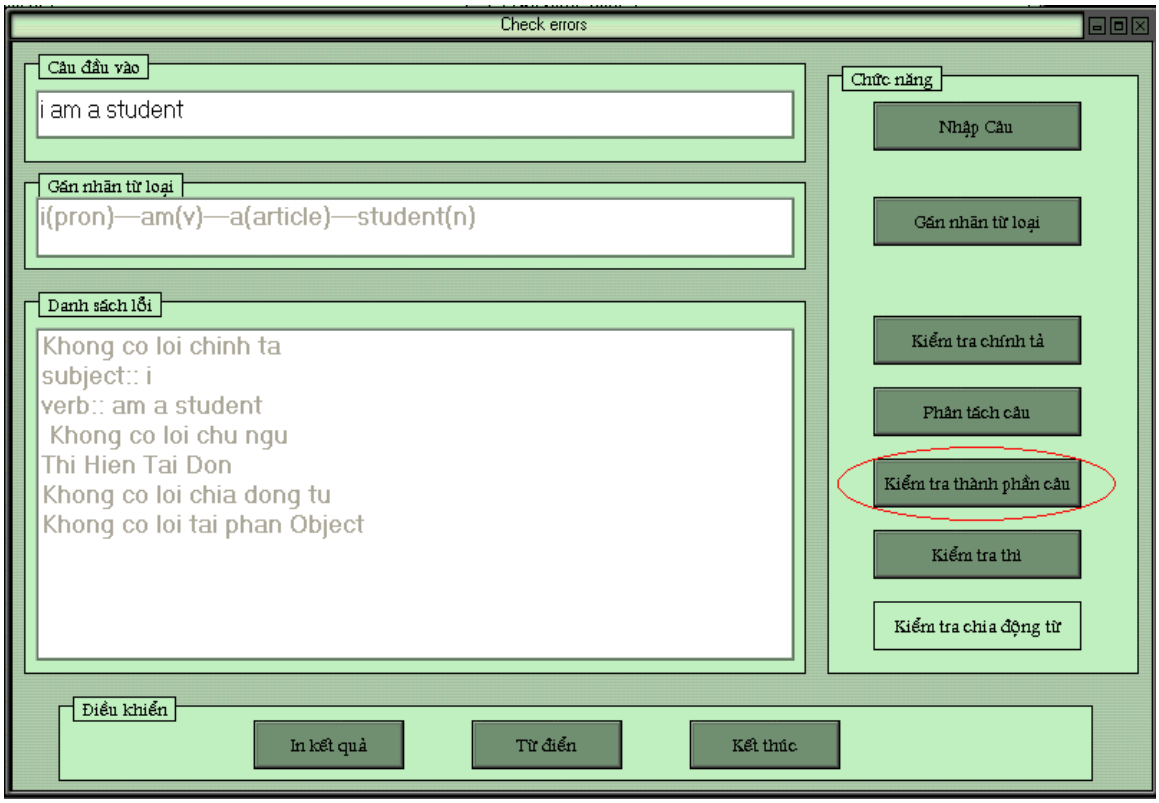
j = UBound(subject)

st = ""

```
For k = 0 To j
   st = st + subject(k) + " "
Next
st = "Subject:: " + st
st = chuanhoa(st)
List1.AddItem st
j = UBound(verb)
st = ""
For k = 0 To j
   st = st + verb(k) + " "
Next
st = "Verb:: " + st
st = chuanhoa(st)
List1.AddItem st
If postrangngu = 0 Then
GoTo a:
Else
j = UBound(trangngu)
st = ""
For k = 0 To j
   st = st + trangngu(k) + " "
Next
st = "ADV:: " + st
st = chuanhoa(st)
List1.AddItem st
End If
a:
Command10.Enabled = True
Command9.Enabled = True
Command8.Enabled = True
End Sub"
```

** Kiểm tra thành phần câu*

Kiểm tra lỗi phần chủ ngữ, trạng ngữ của câu:



"Private Sub Command10_Click()

Dim a As Boolean

If (postrangngu = 1) Or (postrangngu = 3) Then

a = kiemtratrangngu(trangngu)

If a = True Then

List1.AddItem " Khong co loi trang ngu"

Else

List1.AddItem " Co loi trang ngu"

End If

End If

If (postrangngu = 2) Then

a = KTtrangngugiua(trangngu)

If a = True Then

List1.AddItem " Khong co loi trang ngu"

Else

List1.AddItem " Co loi trang ngu"

End If

End If

a = ktsubject(subject)

If a = True Then

List1.AddItem " Khong co loi chu ngu"

Else

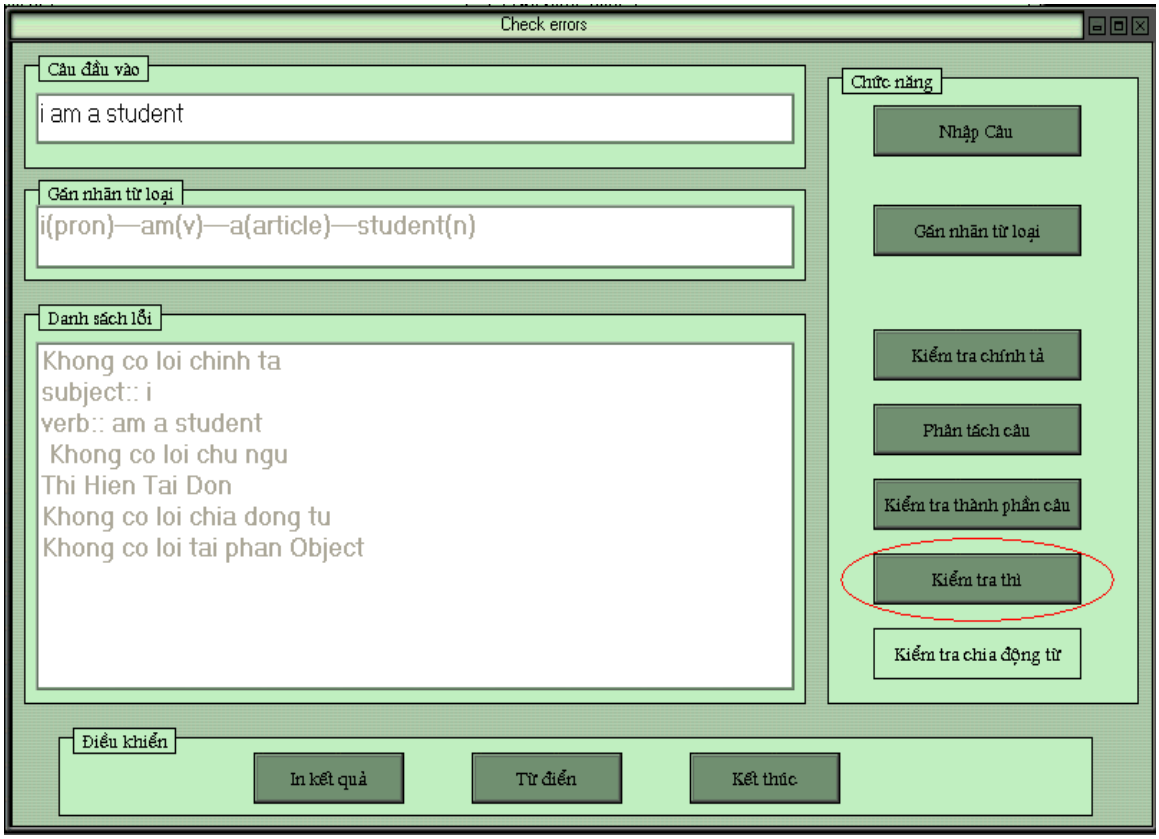
List1.AddItem " Co loi chu ngu"

End If

End Sub"

** Kiểm tra thì*

Kiểm tra câu thuộc thì nào trong số các thì: Hiện tại đơn, hiện tại tiếp diễn, hiện tại hoàn thành, hiện tại hoàn thành tiếp diễn, quá khứ đơn, quá khứ tiếp diễn, tương lai đơn. Nếu không thuộc các thì trên thì thông báo không xác định được thì của câu.



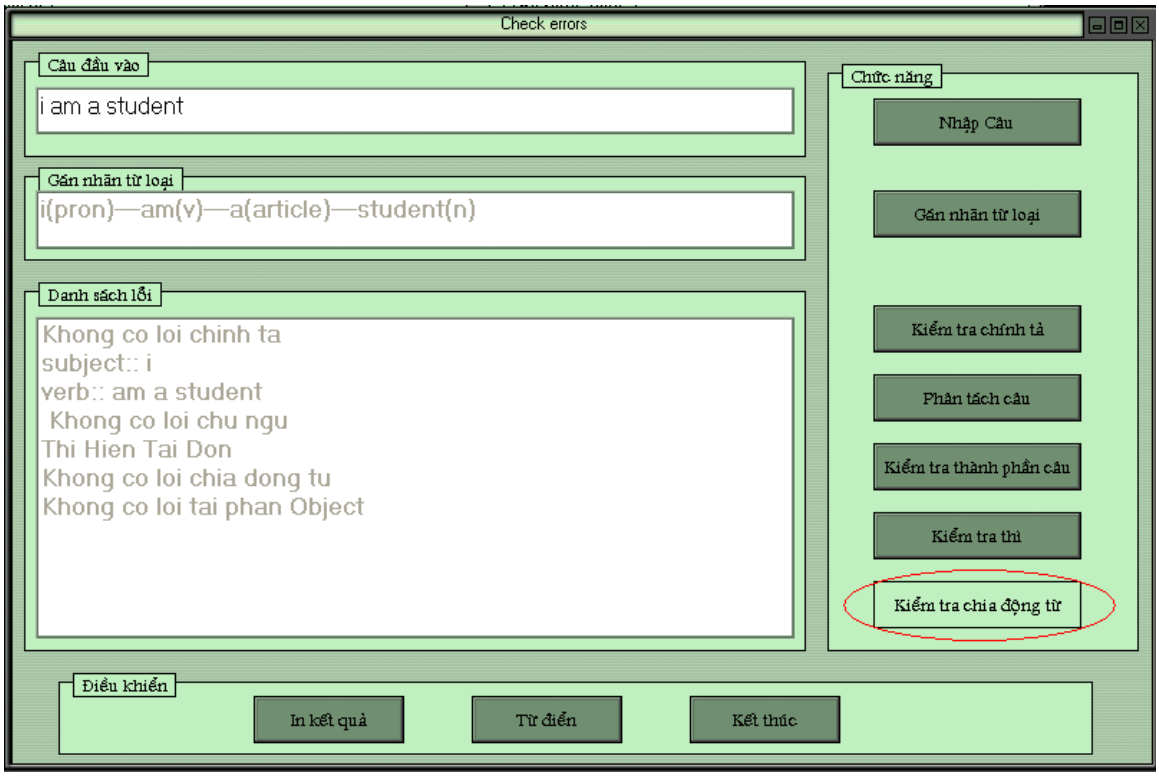
Private Sub Command8_Click()

Dim i As Integer

```
        i = kiemtrathi(verb)
        thi = i
        If thi = 101 Then
           List1.AddItem "Thi Hien Tai Don"
        End If
        If thi = 102 Then
           List1.AddItem "Thi Hien Tai Tiep Dien"
        End If
        If thi = 103 Then
           List1.AddItem "Thi Hien Tai Hoan Thanh"
        End If
        If thi = 104 Then
           List1.AddItem "Thi Hien Tai Hoan Thanh Tiep Dien"
        End If
        If thi = 105 Then
           List1.AddItem "Thi Qua Khu Don"
        End If
        If thi = 106 Then
           List1.AddItem "Thi Qua Khu Tiep Dien"
        End If
        If thi = 107 Then
           List1.AddItem "Thi Tuong Lai Don"
        End If
        If thi = 0 Then
           List1.AddItem "Khong Xac Dinh Duoc Thi Cua Cau"
        End If
    End Sub
```

** Kiểm tra chia động từ*

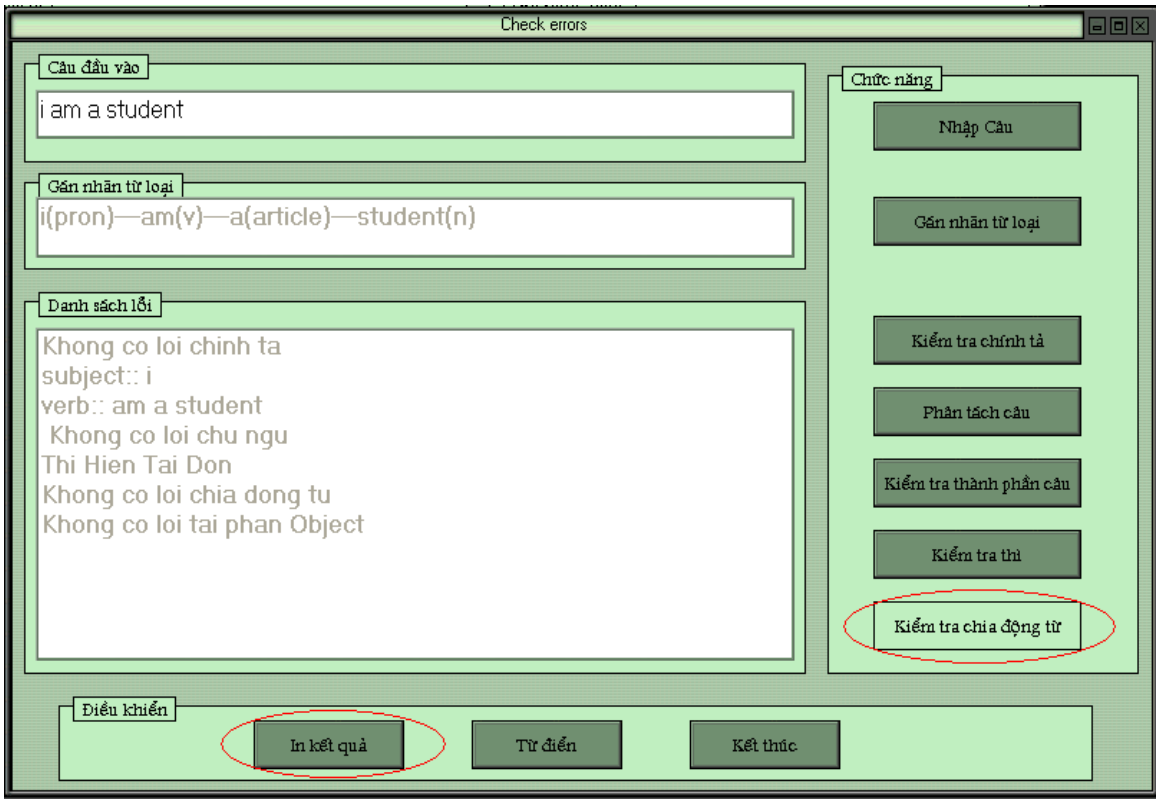
Căn cứ vào việc xác đinh thì của câu mà ta kiểm tra xem có lỗi chia động từ theo đúng thì của câu hay không.




```
Private Sub Command9_Click()
        Dim st As String
        Dim st1 As String
        Dim check As Boolean
        check = kiemtraloitong(subject, verb, st, st1)
        If check = True Then
        List1.AddItem st
        List1.AddItem st1
        End If
End Sub
```

** In kết quả*

In danh sách mã lỗi và lỗi ra file.txt



"Private Sub Command1_Click()

Dim filename As String

Dim i As Integer

Dim temp As String

Dim k As String

Dim kk As String

conn.ShowSave

filename = conn.filename + ".txt"

Open filename For Output As #1

If thi = 101 Then

k = "1"

End If

If thi = 102 Then

k = "2"

End If

If thi = 103 Then

```
k = "3"
End If
If thi = 104 Then
k = "4"
End If
If thi = 105 Then
k = "5"
End If
If thi = 106 Then
k = "6"
End If
If thi = 107 Then
k = "7"
End If
 If thi = 0 Then
k = "0"
End If
 If postrangngu = 0 Then
kk = "0"
End If
   If postrangngu = 1 Then
kk = "1"
End If
 If postrangngu = 2 Then
kk = "2"
End If
 If postrangngu = 3 Then
kk = "3"
End If
Print #1, k
Print #1, kk
```

Print #1, Text1.Text

Print #1, Text2.Text

For i = 0 To List1.ListCount - 1

temp = List1.List(i)

Print #1, List1.List(i)

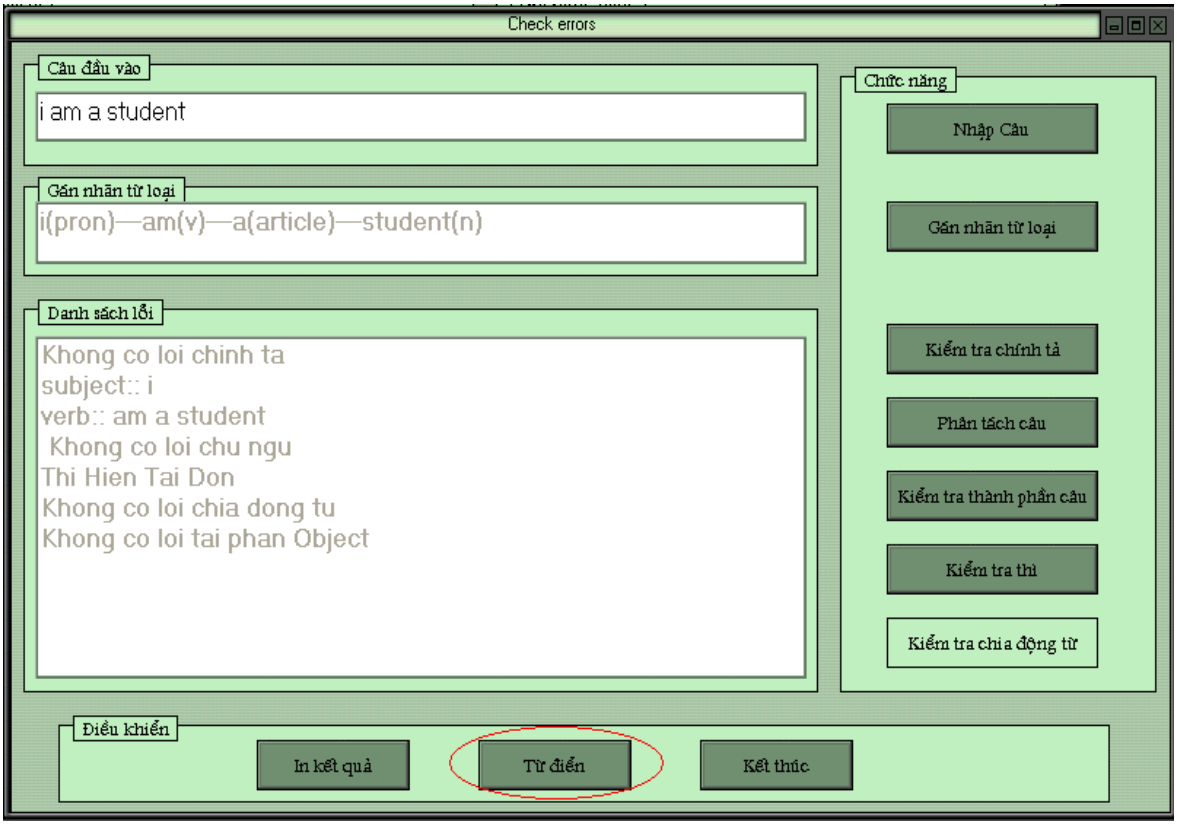
Next

Close #1

End Sub"

** Từ điển*

Gọi đến Form từ điển để làm việc.



"Private Sub Command2_Click()

eng.Show

End Sub"

** Kết thúc*

Kết thúc làm việc với Form.

VI Hạn chế và hướng phát triển của đề tài.

Hạn chế:

+ Chương trình hiện nay chỉ thực hiện được với câu đơn và ở thể khẳng định.

+ Modulo kiểm tra lỗi vị ngữ vẫn chưa được hoàn thiện.

Hướng phát triển:

+ Tiếp tục nghiên cứu để thực hiện cho các loại câu khác để dần hoàn thiện chương trình.

+ Hoàn thiện các modulo tách ghép thành phần câu.

Ứng dụng của đề tài:

+ Làm cơ sở cho các chương trình dịch tự động, chích rút thông tin văn bản, hỗ trợ học tiếng anh trên máy tính...

KẾT LUẬN

Trong quá trình nghiên cứu, tìm hiểu và hoàn thành đồ án tốt nghiệp " Tìm hiểu về xử lý ngôn ngữ tự nhiên và viết chương trình mô phỏng kiểm tra lỗi từ vựng trong việc sử dụng câu tiếng Anh", em đã thu nhận được thêm những kiến thức và em cũng nhận thấy xử lý ngôn ngữ tự nhiên là một lĩnh vực nghiên cứu rộng lớn, còn nhiều điều cần phải khám phá.

Trong đề tài em đã cố gắng tập trung tìm hiểu và nghiên cứu tổng quan về xử lý ngôn ngữ tự nhiên, một số thuật toán phân tích cú pháp và em cũng đã tìm hiểu một số các quy tắc sử dụng từ vựng trong ngôn ngữ tiếng Anh. Từ đó em đã xây dựng được chương trình mô phỏng kiểm tra lỗi từ vựng trong tiếng Anh.

Do thời gian thực hiện đồ án hạn chế nên em mới chỉ tìm hiểu được một số bước trong quá trình xử lý ngôn ngữ tự nhiên và chương trình mô phỏng còn chưa được hoàn thiện như mong muốn. Trong thời gian tới em sẽ cố gắng tiếp tục nghiên cứu và hoàn thiện việc tìm hiểu xử lý ngôn ngữ tự nhiên và chương trình mô phỏng kiểm tra lỗi từ vựng này.

Sinh viên

Đào Văn Trung

TÀI LIỆU THAM KHẢO

[1]. Đinh Điền, Giáo trình xử lý ngôn ngữ tự nhiên, Đại học Khoa Học Tự Nhiên Tp.HCM, 12/2004.

[2]. V.Vapnik, The Nature of Statistical Learning Theory. Springer, NewYork, 1995.

[3]. Allen,J.(1995).Natural Language Understanding. BenjaminCummings,Menlo Park, CA.

[4]. Berger, A. L., Pietra, S. A. D., and Pietra, V. J. D. (1996). Amaximum entropy approach to natural language processingComputational Linguistics, 22(1), 39–71.

[5]. Gazdar, G.andMellish, C.(1989). Natural Language Process-as mental representations of language. In Bresnan, J. (Ed.),ing in LISP. Addison Wesley.

[6]. Mai Lan Hương, Nguyễn Thanh Loan. Ngữ pháp tiếng Anh, Saigonbook, 2007.

[7]. Internet.