TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ**

**HỆ CHUYÊN GIA**

**ĐỀ TÀI:**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **: NGUYỄN HOÀNG MINH**  **NGUYỄN TRỌNG ĐỨC**  **NGUYỄN THỌ HIẾU** |
| **Giảng viên hướng dẫn** | **: VŨ VĂN ĐỊNH** | |
| **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | |
| **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | |
| **Lớp** | **: D13CNPM2** | |
| **Khóa** | **: 2018-2023** | |

**XÂY DỰNG ỨNG DỤNG CHUẨN ĐOÁN BỆNH**

|  |  |
| --- | --- |
| ***Hà Nội, tháng 11 năm 2021*** |  |

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | **Nguyễn Hoàng Minh** |  |  |  |
| 2 | **Nguyễn Trọng Đức** |  |  |  |
| 3 | **Nguyễn Thọ Hiếu** |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

**MỤC LỤC**

Đề tài……………………………………………………………………………….. 1

I. Giới Thiệu Chung ……………………………………………………………….. 2

II. Sơ Lược Cách Biểu Diễn Trong Prolog ………………………………………... 4

III .Kết nối C# với Prolog …………………………………………………………. 5

IV. Cơ Chế Thực Hiện Và Biểu Diễn Luật Trên Ứng Dụng ……………………… 6

V. DEMO Chương Trình ………………………………………………………….. 7

VI. Cơ Chế Làm Việc Nhóm ……………………………………………………… 10

# LỜI MỞ ĐẦU

Thế giới ngày nay phát triển mạnh mẽ với các hoạt động vô cùng đa dạng và phức tạp đòi hỏi khả năng giải quyết vấn đề ở mức độ trí tuệ nhân tạo ngày càng cao . Lĩnh vực trí tuệ nhận tạo nói chung và hệ chuyên gia nói riêng góp phần tạo ra các hệ thống có khả năng trí tuệ của con người, có được tri thức tiên tiến của các hệ chuyên gia để giải quyết các vấn đề phức tạp trong cuộc sống .

Hệ chuyên gia, một nhánh của trí tuệ nhân tạo, là một trong những hướng khai thác tri thức của một ngành thuần lý thuyết, xuất hiện vào thập niên 60 và phần nào chứng minh khả năng áp dụng trí tuệ nhân tạo, khắc phục được các bế tắc gặp phải thời bấy giờ.

Hệ chuyên gia được tiếp cận thảo chương theo một quan niệm mới (phi thủ tục) so với cách tiếp cận thảo chương cổ điển (thủ tục). Với cách tiếp cận này hệ chuyên gia dần dần được đặt vào đúng vị trí vốn có của nó. Ngày nay trong sự phát triển chung của công nghệ thông tin toàn cầu, hệ chuyên gia tiếp tục được phát triển nhằm cung cấp các công cụ hữu hiệu cho những áp dụng tin học vào nhiều ngành nghề khác nhau trong xã hội.

  Trong bước đường khai phá tìm tòi và học hỏi những tri thức mới lạ trong lĩnh vực cơ sở tri thức. Và cũng với giới hạn thời gian không cho phép, cho nên nhóm chúng em chỉ tìm hiểu và cài đặt một ứng dụng nhỏ để chuẩn đoán bệnh sử dụng ngôn ngữ C# kết hợp Prolog. Trong thời gian học tập, tìm hiểu đã gặp nhiều khó khăn và thiếu sót nhưng với sự giúp đỡ nhiệt tình của thầy Vũ Văn Định và các thầy cô trong khoa đã giúp chúng em có những kinh nghiệm quý báu để hoàn thành bài báo cáo này. Em xin chân thành cảm ơn.

**CHƯƠNG 1: TỔNG QUAN VỀ HỆ CHUYÊN GIA**

* 1. **Giới thiệu về hệ chuyên gia**

Trong lĩnh vực trí thông minh nhân tạo, một hệ chuyên gia là một hệ thống máy tính có khả năng mô phỏng hay hành động giống như một chuyên gia thực sự, liên quan đến các vấn đề như dự đoán, đưa ra lời khuyên, hỗ trợ quyết định…

**Sự tương đương của hệ chuyên gia và chuyên gia được thể hiện qua:**

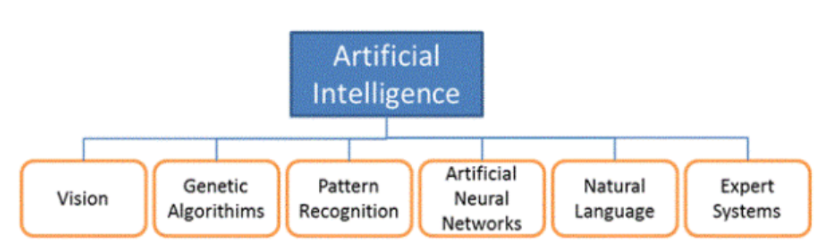
* Khả năng suy luận dựa trên không gian kiến thức của con người.
* Giải quyết các vấn đề phức tạp bằng heuristic hay các kỹ thuật gần đúng.
* Giải thích và điều chỉnh các giải pháp dựa trên các sự kiện.

Các hệ chuyên gia được thiết kế để giải quyết các vấn đề phức tạp bằng cách suy luận trên cơ sở tri thức, và được biểu diễn chủ yếu theo dạng:

IF … THEN … ELSE

Hơn là các thủ tục code tường dùng. Dưới đây là một vài điểm nổi bật của hệ chuyên gia khi so sánh với các chuyên gia khi được phát triển và ứng dụng trong các lĩnh vực.

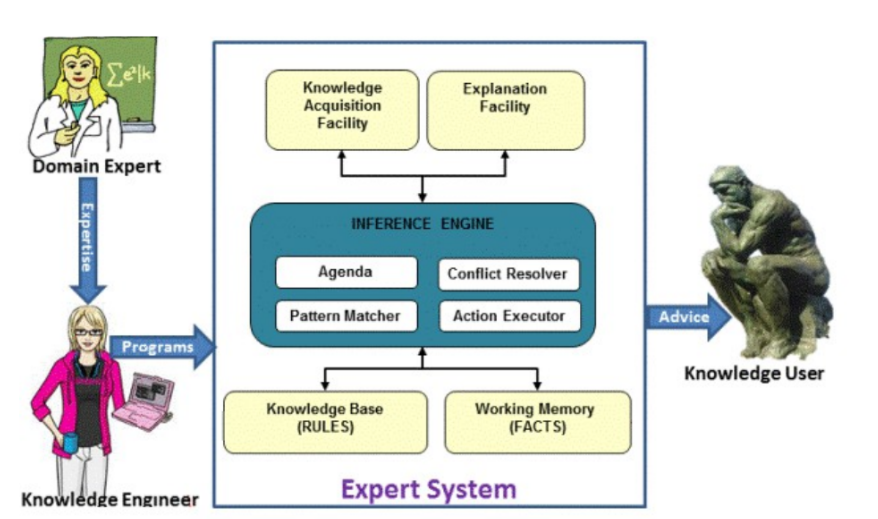
* Chi phí thấp.
* Các kết quả trả về thường là mang lại kết quả đúng.
* Độ tin cậy và tính sẵn sàng cao.
* Thay thế con người trong những môi trường nguy hiểm.
* Sao chép khả năng của một chuyên gia.
* Có khả năng giải thích và suy luận mà không qua cảm xúc, thái độ.



* 1. **Kiến trúc hệ chuyên gia**

Một hệ chuyên gia bao gồm các thành phần sau:

* Bộ hỗ trợ giải thích (**Explanation facility**): cung cấp các kiến thức trong quá trình suy luận khi đưa ra một kết luận hay một quyết định nào đó.
* Bộ hỗ trợ thao tác kiến thức (**Knowledge acquisition facility**): cung cấp phương tiện để nắm bắt các sự kiện và lưu trữ kiến thức được suy luận từ chuyên gia vào trong cơ sở tri thức.
* Cơ sở tri thức (**Knowledge base**): lưu trữ chi thức của con người, thường theo các luật IF … THEN … ELSE.
* Không gian làm việc (**Working memory**): một cơ sở dữ liệu các sự kiện, được sử dụng trong các luật suy diễn.
* Bộ suy diễn (**Inference engine**): thực hiện quá trình suy diễn trên các luật, có thể dựa vào bộ ưu tiên hoặc các kiến thức heuristic để tìm lời giải.
* Bộ ghi nhật ký (**Agenda**): một danh sách các luật đã thỏa mãn theo độ ưu tiên, qua quá trình suy diễn các sự kiện.
* Bộ so khớp mẫu (**Pattern matcher**): so sánh các luật và các sự kiện.



Hình 1.2: Mô hình hệ chuyên gia

**Quá trình thực thi của hệ chuyên gia**

Bộ phận được xem là quan trọng bậc nhất trong hệ chuyên gia là bộ suy diện. Bộ suy diễn hoạt động theo quy trình tròn sau đây:

* Bộ so khớp mẫu (**Pattern matcher**): sử dụng các thuật toán để tạo ra danh sách các luật, dựa vào các sự kiện đã tìm được trong bộ không gia làm việc, ví dụ như thuật toán Rete.
* Bộ ghi nhật ký(**Agenda**): xác định thứ tự mà các luật phát sinh từ danh sách
* Bộ giải quyết mâu thuẫn(**confilct resolver**): chọn các luật có độ ưu tiên các nhất từ bộ ghi nhật ký.
* Bộ thực thi hành động: thực thi các kết quả suy luận vế phải (**Right hand side**) của các luật được chọn và xóa các luật này từ bộ ghi nhật ký.
* Bộ sơ khớp mẫu (**Pattern matcher**): được kích hoạt trở lại, thực hiện cập nhật bộ ghi nhật ký với các luật thỏa mãn vế trái và xóa các luật không thỏa.

Quy trình này tiếp tục cho đến khi không còn luật nào tồn tại trong bộ ghi nhật ký. Một luật được kích hoạt khi và chỉ khi vế trái của nó thỏa mãn. Tính năng quan trọng này trong hệ chuyên gia được gọi là sự khúc xạ (Refraction). Nó đảm bảo rằng một hệ chuyên gia sẽ không bị vòng lặp vô tận.

1.2.1 **Quá trình thu thập tri thức (Knowledge Engineering):**

Thu thập tri thức là một quá trình trích rút dữ liệu của một lĩnh vực cụ thể, thông qua các chuyên gia, sau đó cấu trúc, định dạng và tích hợp vào trong cơ sở tri thức của hệ chuyên gia.

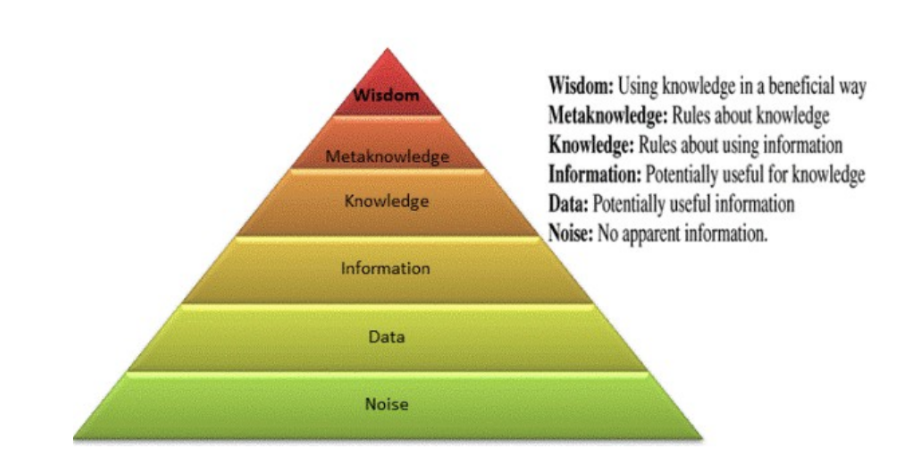
Các dữ liệu rút trích được có thể bao gồm nhiều luật logic hay các kiến thức heuristic được sử dụng bởi các chuyên gia trong một số trường hợp cụ thể nào đó. Không giống như một thuật toán đảm bảo sẽ tìm ra được lời giải chính xác hoàn toàn trong một khoảng thời gian nhất định. Hệ chuyên gia chỉ đảm bảo sẽ tìm được lời giải hợp lý nhất.

Dựa vào trình độ, mức độ kiến thức của chuyên gia trong quá trình hiểu biết sự kết nối giữa các kiến thức hay các suy luận, hệ quả; một ký sư lập trình kiến thức có thể xây dựng nên một hệ chuyên gia sử dụng các quy tắc heuristic. Các kiến thức này thường được biểu diễn dưới dạng các luật được gọi là các cơ sở tri thức.

1.2.1 **Biểu diễn tri thức**

Một chương trình máy tính gồm cấu trúc dữ liệu và các thuật toán. Trong khi đó hệ chuyên gia bao gồm hai phần chính là cơ sở tri thức và bộ suy diễn.

Tri thức là một phần trong mô hình phân cấp “Tháp kiến thức”:



Hình 1.3: Tháp tri thức

* Dữ liệu tạp (Noise): các loại dữ liệu không xác định, không rõ rang
* Data: Những dữ liệu hữu ích, có thể dùng để trích xuất các thông tin
* Thông tin (Information): những dữ liệu hữu ích cho kiến thức
* Kiến thức (Knowledge): các luật rút ra từ các thông tin
* Siêu tri thức (Metaknowledge): các luật rút ra từ các tri thức

1.3 **Các ứng dụng của hệ chuyên gia**

Hayes-Roth chia các ứng dụng của hệ chuyên gia thành 10 loại:





1.4 **Một số hạn chế của hệ chuyên gia**

- Hệ chuyên gia có thể hoạt động tốt trong một lĩnh vực hẹp với một độ phức tạp nhất định. Việc mô hình hóa một hệ chuyên gia cho nhiều lĩnh vực có thể gây tác dụng ngược lại.

- Hệ chuyên gia không hoàn toàn đảm bảo chất lượng hay tính đúng đắn của các luật trong quá trình suy luận

- Việc thêm các luật, các tri thức mới có thể sẽ xung đột với những luật hiện tại và có thể giảm độ tin cậy của hệ thống.

**CHƯƠNG 2: XÂY DỰNG VÀ PHÂN TÍCH BÀI TOÁN**

**2.1. Giới thiệu bài toán**

Bệnh lý là điều chắc chắn mỗi người đều gặp phải, nó xảy ra thường xuyên tùy mức độ tùy người, nên mỗi người chúng ta cần nắm bắt một số kiến thức cơ bản về các bệnh thường gặp như cảm sốt, mào gà, ho, việm họng … để có các biện pháp kịp thời hoặc chúng ta cũng biết được liệu mình có bị mắc bệnh gì không khi không đến gặp bác sĩ được hoặc chúng ta không muốn, đó là điều hết sức quan trọng.

Chính vì thế ứng dụng này của chúng em phục vụ cho mục đích đó, ứng dụng cho ra kết của cụ thể tên bệnh là gì? Giải thích nguyên nhân gây bệnh, từ những biểu hiện gặp phải mà người dùng chọn máy có thể suy đoán dựa trên các tập luật đã được định nghĩa sẵn trong chương trình. Bộ luật này được tổ chức và thực thi trên ngôn ngữ Prolog , kết hợp với lệnh của C# và giao diện trực quan giúp người bình thường có thể dễ dàng sử dụng.

Ứng dụng sẽ đưa ra tên bệnh cụ thể bằng ngôn ngữ tự nhiên, giúp người dùng có thể hiểu, và đặc biệt có phần giải thích nguyên nhân gây bệnh thế nào, từ các biểu hiện đó máy suy nghĩ và đưa ra kết luận hợp lý nhất.

Hướng dẫn sử dụng:  
+ Vì đây là ứng dụng giao diện nên rất dễ để sử dụng; đầu tiên chạy file .exe lên

+ Chương trình sẽ kích hoạt cửa sổ và người dùng làm theo những gì được ghi trực tiếp trên ứng dụng

+ Phần kết quả sẽ được hiển thị cuối cùng.

+ Người dùng xem kết quả và giải thích

Nhưng cũng phải thừa nhận rằng, đây chỉ là ứng dụng với mục đích thực hiện đồ án môn học, để thực hành và củng cố kiến thức nên độ chính xác áp dụng trong thực tế thì chưa cao. Và nhất là đây là ứng dụng dự đoán bệnh của con người, liên quan đến sức khỏe và cơ thể sinh học nên chắc chắn còn nhiều thiếu xót, vì thế nhóm sẽ cố gắng để hoàn thiện đồ án hơn nữa.

1. **SƠ LƯỢC CÁCH BIỂU DIỄN CÁC TẬP LUẬT TRÊN PROLOG**

Các tập luật được biểu diễn trên Prolog theo dạng:

**Dữ kiện(Bệnh, tên bệnh):- Các triệu chứng của bệnh.**

**Ví dụ:**

giathiet(Benh,soi) :- // Giả thiết đây là bệnh sởi, thì

trieuchung(Benh,sot), luật bao gồm những triệu chứng của sởi.

trieuchung(Benh,ho),

trieuchung(Benh,dauhong),

trieuchung(Benh,somui),

trieuchung(Benh,noido).

**Tương tự cho những luật ở dưới.**

giathiet(Benh,cum) :-

trieuchung(Benh,sot),

trieuchung(Benh,daudau),

trieuchung(Benh,nhucmoi),

trieuchung(Benh,dauhong),

trieuchung(Benh,onlanh),

trieuchung(Benh,chaymui),

trieuchung(Benh,somui),

trieuchung(Benh,ho).

giathiet(Benh,camthuong) :-

trieuchung(Benh,daudau),

trieuchung(Benh,hatxi),

trieuchung(Benh,chaymui),

trieuchung(Benh,somui),

trieuchung(Benh,onlanh).

giathiet(Benh,phatban) :-

trieuchung(Benh,noido),

trieuchung(Benh,sot),

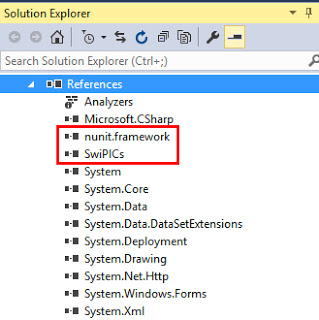
trieuchung(Benh,nhucmoi).

1. **KẾT NỐI C# VỚI PROLOG**

Kết nối C# với Prolog cần có:

* Prolog phiên bản 32 bit 6.6.5
* SwiPlCs và nunit.framework.dll: Thư viện kết nối C# và Prolog phiên bản 1.1.60605.0
* Visual Studio 2013

Ta sẽ thêm thư viện SwiPlCs vào project bằng cách từ giao diện project của Visual Studio chọn chuột phải vào **References --> Add references**.... Sau đó chọn **Browse...** rồi trỏ tới thư mục **Debug** của project và thêm 2 file ***SwiPlCs.dll*** và ***nunit.framework.dll*** vào **References**



**Nhưng trong trường hợp này, nhóm em sẽ gán cứng luật vào trong chương trình. Và sẽ không cần tới sự lằng nhằng khi kết nối C# với prolog nữa. Nhằm sử dụng chương trình một cách dễ dàng và tránh các khả năng lỗi.**

1. **CƠ CHẾ THỰC HIỆN VÀ BIỂU DIỄN LUẬT TRÊN ỨNG DỤNG**

CƠ CHẾ THỰC HIỆN:

Chương trình sẽ đưa ra những câu hỏi và câu trả lời theo dạng **CÓ** hoặc **KHÔNG**. Và những gì người dùng cần làm là chọn và trả lời.

BIỂU DIỄN LUẬT:

**e1** Nổi mẩn đỏ khắp người. **e6** Cảm thấy ớn lạnh

**e2** Đau đầu **e7** Bị đau họng

**e3** Sổ mũi **e5** Nhức mỏi cơ thể

**e4** Sốt **e8** Bị ho

**e9** Hắt xì hơi liên tục

**c10** Bạn bị sởi rồi! **c12** Bạn bị cảm lạnh!

**c11** Bạn bị cảm cúm rồi! **c13** Bạn bị sốt phát ban!

Ta có mỗi luật ( r ) là tập hợp của các giả thiết ( e ) đưa ra, và xâu chuỗi luật lại sẽ đưa ra được kết quả ( c ). Dấu & biểu diễn là phép nối, dấu ~ biểu diễn phép ngược giả thiết ( hay còn gọi là **Không** ).

**Ví dụ:**

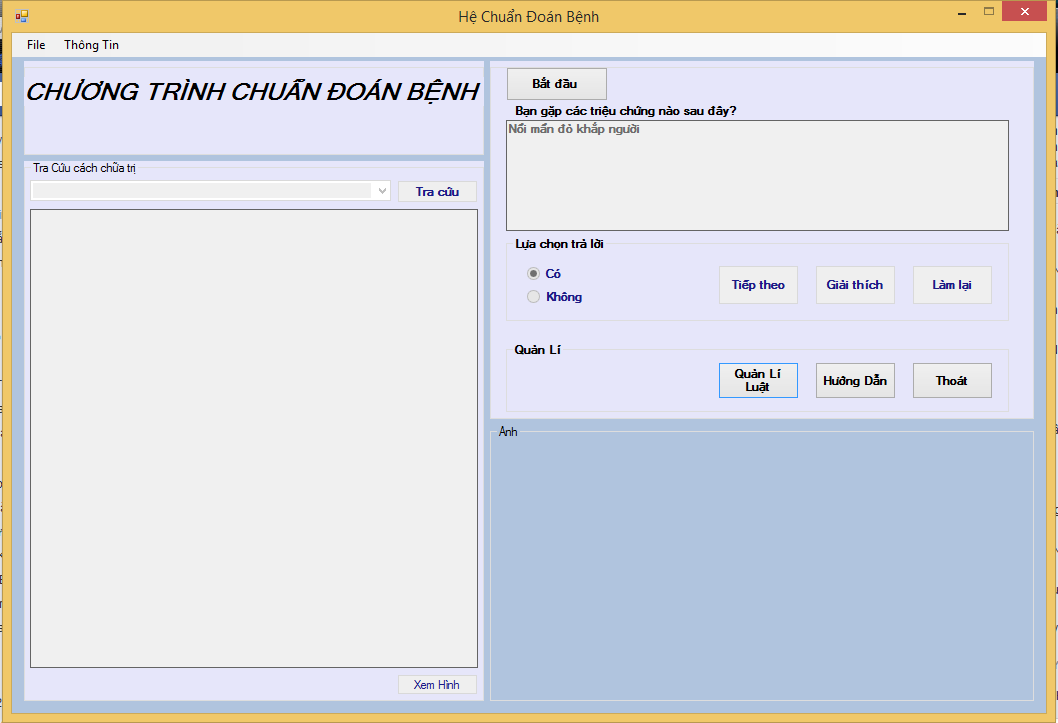
r1 e1&e2&e3&e4=>c10

Luật r1 biểu diễn các giả thiết gồm e1&e2&e3&e4 sẽ cho kết quả là c10.

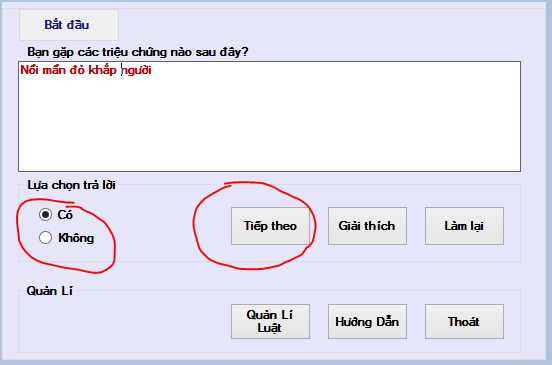
**CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH**

1. **DEMO CHƯƠNG TRÌNH**

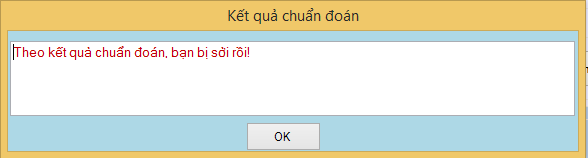
Chương trình được viết trên công cụ Visual Studio 2013.



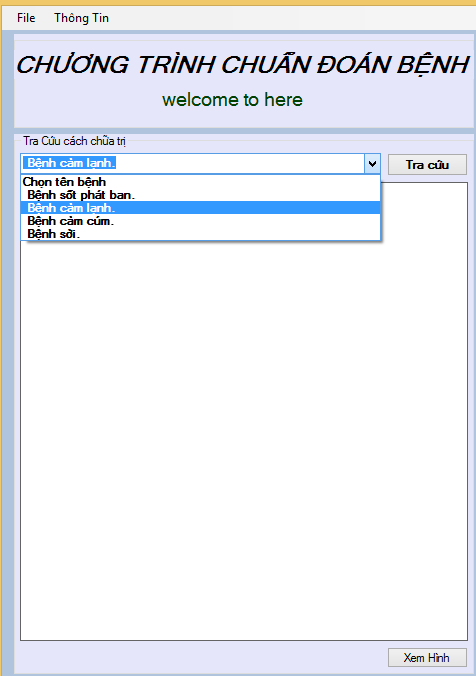
B1: Nhấn Bắt đầu để chương làm việc.



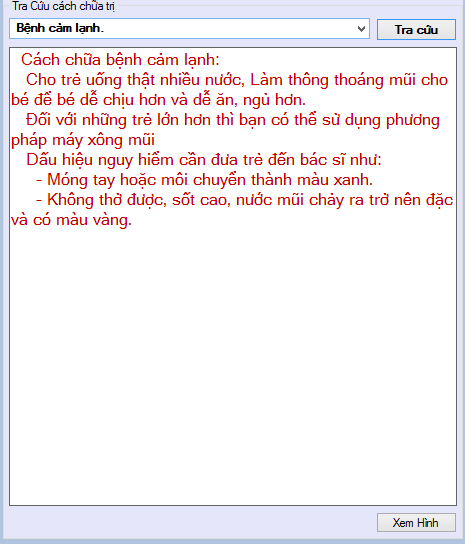
Chọn Có hoặc Không rồi nhấn Tiếp Theo để trả lời các câu hỏi và đi tới câu hỏi khác.



Khi câu hỏi cuối cùng đã trả lời xong, chương trình sẽ đưa ra kết quả!



Chọn Tra cứu cách chữa bệnh, chọn tên bệnh và nhấn Tra Cứu để tìm cách phòng và chữa bệnh.



Kết quả cách chữa trị.

Sau khi sử dụng chương trình, nhấn Thoát để thoát chương trình.

1. **CƠ CHẾ LÀM VIỆC NHÓM**

|  |  |
| --- | --- |
| **Bùi khoa hoàng** | Thiết kế giao diện và |
| **Phạm văn lực** | Tìm kiếm tài liệu |
| **Hà huy hoàng** | Tạo luật trên prolog và |
| **Trần tiến dũng** | Xây dựng ứng dụng. |

* **Hoạt động của nhóm:**
* Thảo luận trực tiếp thông qua học nhóm tại địa điểm nhất định khi có đủ thành viên (cụ thể tại trường ĐH CNTT TPHCM).
* Thảo luận bàn giao công việc chi tiết thông qua các công cụ chát như: skype, facebook, xử lý trực tiếp thông qua teamview, vv…
* **Đánh giá thành viên:**

Tất cả thành viên đều hoàn thành công việc cụ thể đã phân công.

**TÀI LIỆU THAM KHẢO**

<https://nguyenvanhieu.vn/huong-dan-prolog/>

<https://tailieu.vn/doc/bai-giang-he-chuyen-gia-chuong-1-gioi-thieu-he-chuyen-gia-1924685.html>

<https://www.academia.edu/30165943/H%E1%BB%87_chuy%C3%AAn_gia_Nh%C3%B3m>