

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC
CƠ SỞ TRI THỨC VÀ ỨNG DỤNG**

**ĐỀ TÀI:
XÂY DỰNG ỨNG DỤNG HỌC LẬP
TRÌNH C**

**Giảng viên hướng dẫn:
Học viên:**

**PGS. TS. Đỗ Văn Nhơn
Nguyễn Trung Hiếu – CH1901020**

Mục lục

I.	Giới thiệu	2
1.	Tổng quan về Ontology	2
2.	Các thành phần trong ontology.....	2
2.1	Các cá thể (Individuals) - Thể hiện.....	2
2.2	Các lớp (Classes) - Khái niệm.....	3
2.3	Các thuộc tính (Properties)	3
2.4	Các mối quan hệ (Relation).....	3
II.	Tổ chức tri thức kiến thức lập trình C.....	3
1.	Cấu trúc.....	4
1.1	Lớp (Classes):	4
1.2	Thuộc tính (Properties)	9
1.3	Đối tượng (Individual)	18
1.4	Quan hệ (Relations)	20
2.	Giới thiệu ứng dụng Protégé	23
III.	Kết luận	24
IV.	Tài liệu tham khảo	24

I. Giới thiệu

Lập trình C là một ngôn ngữ cơ bản và mạnh mẽ, đã định hình nên nền tảng cho nhiều hệ thống và ứng dụng công nghệ thông tin quan trọng. Tuy nhiên, quá trình hiểu và quản lý tri thức liên quan đến lập trình C không phải lúc nào cũng dễ dàng. Để tiếp cận với ngôn ngữ này và tạo ra mô hình tri thức cấu trúc, sử dụng ontology đã trở thành một công cụ quan trọng và hiệu quả để biểu diễn tri thức lập trình C.

1. Tổng quan về Ontology

Trong khoa học máy tính, một ontology là một mô hình dữ liệu biểu diễn một lĩnh vực và được sử dụng để suy luận về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng. Ontology cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này. Ngoài bộ từ vựng, ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng, nó được sử dụng trong một miền mà có thể được giao tiếp giữa người và các hệ thống ứng dụng phân tán hỗn tạp khác.

Các ontology được sử dụng như là một biểu mẫu trình bày tri thức về thế giới hay một phần của nó. Các ontology thường miêu tả:

Các cá thể: Các đối tượng cơ bản, nền tảng

Các lớp: Các tập hợp, hay kiểu của các đối tượng

Các thuộc tính: Thuộc tính, tính năng, đặc điểm, tính cách, hay các thông số mà các đối tượng có và có thể đem ra chia sẻ.

Các mối liên hệ: Các con đường mà các đối tượng có thể liên hệ tới một đối tượng khác.

Bộ từ vựng ontology được xây dựng trên cơ sở tầng RDF và RDFS, cung cấp khả năng biểu diễn ngữ nghĩa mềm dẻo cho tài nguyên Web và có khả năng hỗ trợ lập luận.

2. Các thành phần trong ontology

2.1 Các cá thể (Individuals) - Thể hiện

Các cá thể là các thành phần cơ bản, nền tảng của một ontology. Các cá thể trong một ontology có thể bao gồm các đối tượng cụ thể như con người, động vật, cái bàn... cũng như các cá thể trừu tượng như các thành viên hay các từ. Một ontology có thể không cần bất kỳ một cá thể nào, nhưng một trong những lý do chính của một ontology là để cung cấp một ngữ nghĩa của việc phân lớp các cá thể, mặc dù các cá thể này không thực sự là một phần của ontology.

2.2 Các lớp (Classes) - Khái niệm

Các lớp là các nhóm, tập hợp các đối tượng trừu tượng. Chúng có thể chứa các cá thể, các lớp khác, hay là sự phối hợp của cả hai. Các ontology biến đổi tùy thuộc vào cấu trúc và nội dung của nó: Một lớp có thể chứa các lớp con, có thể là một lớp tổng quan (chứa tất cả mọi thứ), có thể là lớp chỉ chứa những cá thể riêng lẻ, Một lớp có thể xếp gộp vào hoặc bị xếp gộp vào bởi các lớp khác. Mỗi quan hệ xếp gộp này được sử dụng để tạo ra một cấu trúc có thứ bậc các lớp, thường là với một lớp thông dụng nhất kiểu Thing ở trên đỉnh và các lớp rất rõ ràng kiểu 2002, Ford ở phía dưới cùng.

2.3 Các thuộc tính (Properties)

Các đối tượng trong ontology có thể được mô tả thông qua việc khai báo các thuộc tính của chúng. Mỗi một thuộc tính đều có tên và giá trị của thuộc tính đó. Các thuộc tính được sử dụng để lưu trữ các thông tin mà đối tượng có thể có. Ví dụ, đối với một cá nhân có thể có các thuộc tính: Họ tên, ngày sinh, quê quán, số cmnd... Giá trị của một thuộc tính có thể có các kiểu dữ liệu phức tạp.

2.4 Các mối quan hệ (Relation)

Một trong những ứng dụng quan trọng của việc sử dụng các thuộc tính là để mô tả mối liên hệ giữa các đối tượng trong ontology. Một mối quan hệ là một thuộc tính có giá trị là một đối tượng nào đó trong ontology. Một kiểu quan hệ quan trọng là kiểu quan hệ xếp gộp (subsumption). Kiểu quan hệ này mô tả các đối tượng nào là các thành viên của các lớp nào của các đối tượng. Hiện tại, việc kết hợp các ontology là một tiến trình được làm phần lớn là thủ công, do vậy rất tốn thời gian và đắt đỏ. Việc sử dụng các ontology là cơ sở để cung cấp một định nghĩa thông dụng của các thuật ngữ cốt lõi có thể làm cho tiến trình này trở nên dễ quản lý hơn. Hiện đang có các nghiên cứu dựa trên các kỹ thuật sản sinh để nối kết các ontology, tuy nhiên lĩnh vực này mới chỉ hiện hữu về mặt lý thuyết.

II. Tổ chức tri thức kiến thức lập trình C

1. Cấu trúc

1.1 Lớp (Classes):

Lớp biểu diễn các khái niệm lập trình C như biến, hàm, câu lệnh, kiểu dữ liệu, cấu trúc dữ liệu, con trỏ. Mỗi lớp có một tên và mô tả.

- Lớp Biến (Variable)

Biến là một vùng trong bộ nhớ dùng để lưu trữ dữ liệu. Nó có kiểu, tên biến và giá trị.

Thuộc tính: Kiểu dữ liệu, tên biến, giá trị, phạm vi sử dụng.

Lớp con:

+ Lớp biến có tham số (Parameter Variable):

Là một lớp con của "Biến" và đại diện cho các biến được sử dụng như tham số trong các hàm hoặc phương thức.

Thuộc Tính Bổ Sung: Vị trí tham số, tên tham số.



- Lớp hàm (Function)

Hàm là một khối mã thực hiện một tác vụ cụ thể và có thể được gọi từ chương trình chính hoặc từ các hàm khác.

Thuộc Tính: Tên hàm, kiểu trả về, tham số đầu vào.

Lớp con:

+ Lớp hàm Gọi Đệ Quy (Recursive Function)

Là một lớp con của "Hàm" và đại diện cho các hàm được gọi đệ quy, nghĩa là hàm gọi chính nó.

Thuộc Tính Bổ Sung: Điều kiện dừng, số lần gọi đệ quy.

+ Lớp hàm Thư Viện (Library Function)

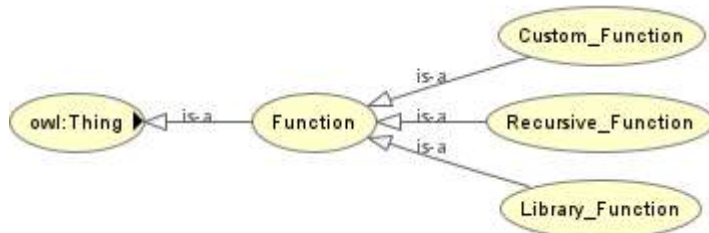
Là một lớp con của "Hàm" và đại diện cho các hàm thư viện có sẵn trong lập trình C.

Thuộc Tính Bổ Sung: Tên hàm thư viện, mô tả chức năng.

+ Lớp hàm Tùy Chỉnh (Custom Function)

Là một lớp con của "Hàm" và đại diện cho các hàm do người dùng tự định nghĩa.

Thuộc Tính Bổ Sung: Mô tả chức năng, tham số cụ thể.



- Lớp kiểu dữ liệu (Data Type)

Kiểu dữ liệu xác định loại dữ liệu mà một biến có thể chứa, ví dụ: int, float, char.

Thuộc Tính: Tên kiểu dữ liệu, kích thước, miêu tả.

Lớp con:

+ Lớp kiểu Dữ Liệu Tùy Chỉnh (Custom Data Type)

Là một lớp con của "Kiểu Dữ Liệu" và đại diện cho các kiểu dữ liệu do người dùng tự định nghĩa trong lập trình C.

Thuộc Tính Bổ Sung: Mô tả, cấu trúc dữ liệu.

+ Lớp kiểu Dữ Liệu Con Trỏ (Pointer Data Type):

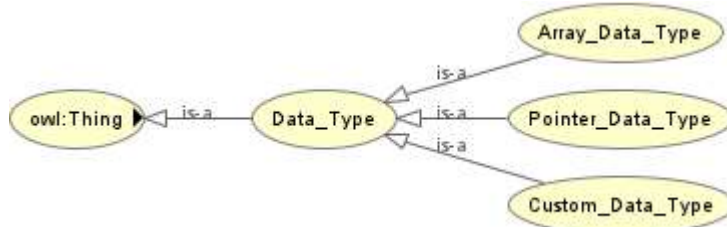
Là một lớp con của "Kiểu Dữ Liệu" và đại diện cho các kiểu dữ liệu con trỏ như int*, float*, char*.

Thuộc Tính Bổ Sung: Điều hướng con trỏ.

+ Lớp kiểu Dữ Liệu Mảng (Array Data Type)

Là một lớp con của "Kiểu Dữ Liệu" và đại diện cho các kiểu dữ liệu mảng như int[], float[], char[].

Thuộc Tính Bổ Sung: Kích thước mảng, số lượng phần tử.



- Lớp điều kiện (Conditional Statement)

Lệnh điều kiện cho phép kiểm tra điều kiện và thực hiện các hành động khác nhau dựa trên kết quả của kiểm tra.

Thuộc Tính: Loại lệnh điều kiện, điều kiện kiểm tra.

Lớp con:

+ Lớp "Lệnh Điều Kiện IF (If Statement)":

Mô tả: Là một lớp con của "Lệnh Điều Kiện" và đại diện cho lệnh điều kiện "if" trong lập trình C.

Thuộc Tính Bổ Sung: Biểu thức điều kiện, khối mã thực thi nếu điều kiện đúng.

+ Lớp "Lệnh Điều Kiện ELSE (Else Statement)":

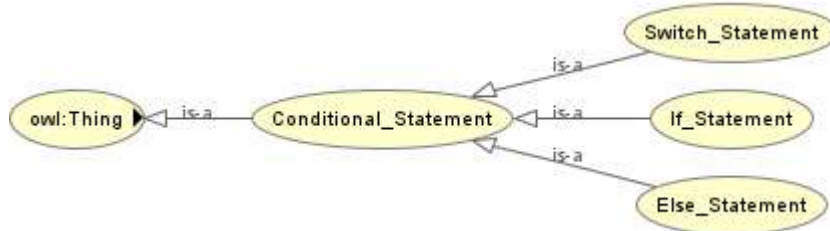
Mô tả: Là một lớp con của "Lệnh Điều Kiện" và đại diện cho lệnh điều kiện "else" trong lập trình C.

Thuộc Tính Bổ Sung: Khối mã thực thi nếu điều kiện "if" sai.

+ Lớp "Lệnh Điều Kiện SWITCH (Switch Statement)":

Mô tả: Là một lớp con của "Lệnh Điều Kiện" và đại diện cho lệnh điều kiện "switch" trong lập trình C.

Thuộc Tính Bổ Sung: Giá trị chuyển đổi, các trường hợp và mã thực thi tương ứng.



- Lớp vòng lặp (Loop)

Vòng lặp cho phép lặp lại một khối mã nhiều lần cho đến khi điều kiện dừng được thoả mãn.

Thuộc Tính: Loại vòng lặp, điều kiện dừng, khối lặp.

Lớp con:

+ Lớp "Vòng Lặp FOR (For Loop)":

Mô tả: Là một lớp con của "Vòng Lặp" và đại diện cho vòng lặp "for" trong lập trình C.

Thuộc Tính Bổ Sung: Biểu thức khởi đầu, điều kiện lặp, bước nhảy và khối mã thực thi.

+ Lớp "Vòng Lặp WHILE (While Loop)":

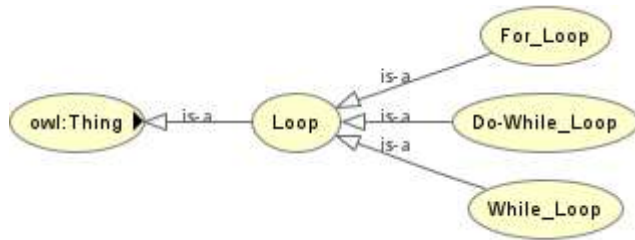
Mô tả: Là một lớp con của "Vòng Lặp" và đại diện cho vòng lặp "while" trong lập trình C.

Thuộc Tính Bổ Sung: Biểu thức điều kiện và khối mã thực thi.

+ Lớp "Vòng Lặp DO-WHILE (Do-While Loop)":

Mô tả: Là một lớp con của "Vòng Lặp" và đại diện cho vòng lặp "do-while" trong lập trình C.

Thuộc Tính Bổ Sung: Biểu thức điều kiện và khối mã thực thi.



Lớp con trỏ (Pointer)

Con trỏ là một biến đặc biệt chứa địa chỉ của biến hoặc hàm khác trong bộ nhớ.

Thuộc Tính: Kiểu con trỏ, địa chỉ trỏ tới.

Lớp con:

+ Lớp con Trỏ Số Nguyên (Integer Pointer)

Là một lớp con của "Con Trỏ" và đại diện cho các con trỏ có kiểu `int*`.

Thuộc Tính Bổ Sung: Kiểu dữ liệu con trỏ, địa chỉ trỏ tới.

+ Lớp Con Trỏ Số Thực (Float Pointer)

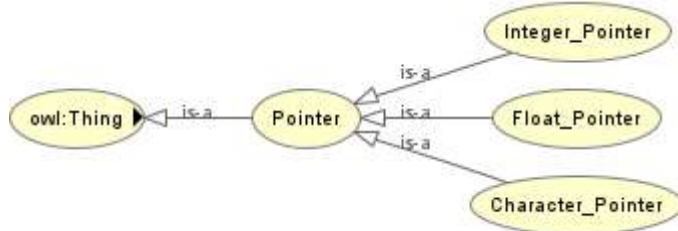
Là một lớp con của "Con Trỏ" và đại diện cho các con trỏ có kiểu `float*`.

Thuộc Tính Bổ Sung: Kiểu dữ liệu con trỏ, địa chỉ trỏ tới.

Lớp con Trỏ Ký Tự (Character Pointer):

Mô tả: Là một lớp con của "Con Trỏ" và đại diện cho các con trỏ có kiểu `char*`.

Thuộc Tính Bổ Sung: Kiểu dữ liệu con trỏ, địa chỉ trỏ tới.



- Lớp mảng (Array)

Mảng là một tập hợp các biến có cùng kiểu dữ liệu được lưu trữ liên tiếp trong bộ nhớ.

Thuộc Tính: Kiểu dữ liệu của mảng, kích thước, phần tử.

Lớp con:

+ Lớp mảng Số Nguyên (Integer Array)

Mô tả: Là một lớp con của "Mảng" và đại diện cho các mảng có kiểu dữ liệu `int[]`.

Thuộc Tính Bổ Sung: Kiểu dữ liệu của mảng, kích thước, giá trị các phần tử.

+ Lớp mảng Số Thực (Float Array)

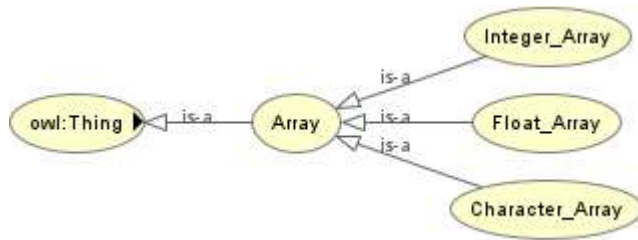
Mô tả: Là một lớp con của "Mảng" và đại diện cho các mảng có kiểu dữ liệu `float[]`.

Thuộc Tính Bổ Sung: Kiểu dữ liệu của mảng, kích thước, giá trị các phần tử.

+ Lớp mảng Ký Tự (Character Array)

Mô tả: Là một lớp con của "Mảng" và đại diện cho các mảng có kiểu dữ liệu char[].

Thuộc Tính Bổ Sung: Kiểu dữ liệu của mảng, kích thước, giá trị các phần tử.



Lớp Thư viện (Library)

Thư viện là một tập hợp các hàm và tài liệu liên quan được sử dụng để phát triển chương trình.

Thuộc Tính: Tên thư viện, hàm có sẵn.



- Lớp cấu trúc dữ liệu (Data Structure)

Cấu trúc dữ liệu như struct và union được sử dụng để tổ chức dữ liệu trong lập trình C.

C.

Thuộc Tính: Loại cấu trúc dữ liệu, các thành phần dữ liệu.

Lớp con:

+ Lớp cấu Trúc Dữ Liệu Danh Sách Liên Kết (Linked List)

Mô tả: Là một lớp con của "Cấu Trúc Dữ Liệu" và đại diện cho cấu trúc dữ liệu danh sách liên kết trong lập trình C.

Thuộc Tính Bổ Sung: Loại danh sách liên kết (singly linked list, doubly linked list), các phần tử và liên kết giữa chúng.

+ Lớp cấu Trúc Dữ Liệu Ngăn Xếp (Stack):

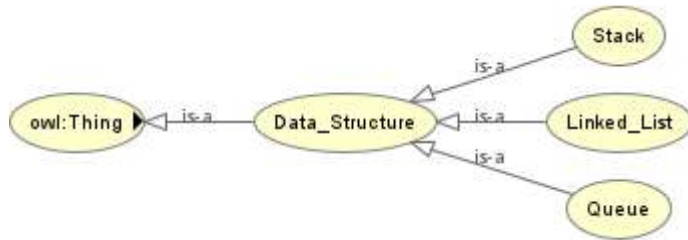
Mô tả: Là một lớp con của "Cấu Trúc Dữ Liệu" và đại diện cho cấu trúc dữ liệu ngăn xếp trong lập trình C.

Thuộc Tính Bổ Sung: Loại ngăn xếp, các phần tử và quy tắc hoạt động.

+ Lớp cấu Trúc Dữ Liệu Hàng Đợi (Queue):

Mô tả: Là một lớp con của "Cấu Trúc Dữ Liệu" và đại diện cho cấu trúc dữ liệu hàng đợi trong lập trình C.

Thuộc Tính Bổ Sung: Loại hàng đợi (queue), các phần tử và quy tắc hoạt động.



- Lớp ghi chú (Comment)

Ghi chú là các phần chú thích trong mã nguồn giúp người đọc hiểu mã nguồn hơn.

Thuộc Tính: Nội dung ghi chú, vị trí.

Lớp con:

+ Lớp ghi Chú Mô Tả (Description Comment)

Mô tả: Là một lớp con của "Ghi Chú" và đại diện cho các ghi chú được sử dụng để mô tả chức năng hoặc ý nghĩa của mã nguồn.

Thuộc Tính Bổ Sung: Nội dung mô tả.

+ Lớp ghi Chú Làm Chú Ý (TODO Comment)

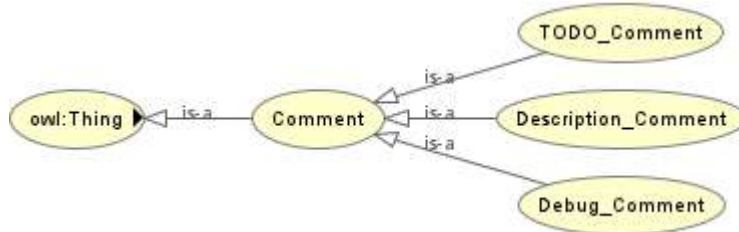
Mô tả: Là một lớp con của "Ghi Chú" và đại diện cho các ghi chú được sử dụng để đánh dấu các nhiệm vụ hoặc công việc cần thực hiện trong tương lai.

Thuộc Tính Bổ Sung: Nội dung công việc cần thực hiện.

+ Lớp ghi Chú Gỡ Lỗi (Debug Comment)

Mô tả: Là một lớp con của "Ghi Chú" và đại diện cho các ghi chú được sử dụng để ghi lại thông tin về gỡ lỗi mã nguồn.

Thuộc Tính Bổ Sung: Nội dung thông tin gỡ lỗi.



1.2 Thuộc tính (Properties)

Thuộc tính được gán cho các lớp để mô tả các đặc điểm của chúng.

- Kiểu dữ liệu
:Kiểu_dữ_liệu rdf:type owl:DatatypeProperty ;
rdfs:domain :Biến ;

```
    rdfs:range xsd:string ;  
    rdfs:label "Kiểu dữ liệu" .
```

- Tên biến

```
:Tên rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Biến ;  
    rdfs:range xsd:string ;  
    rdfs:label "Tên biến" .
```

- Giá trị

```
:Giá_trị rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Biến ;  
    rdfs:range xsd:string ;  
    rdfs:label "Giá trị" .
```

- Phạm vi sử dụng

```
:Phạm_vi_sử_dụng rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Biến ;  
    rdfs:range xsd:string ;  
    rdfs:label "Phạm vi sử dụng" .
```

- Vị trí tham số

```
:Vị_trí_tham_số rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Tham_số ;  
    rdfs:range xsd:int ;  
    rdfs:label "Vị trí tham số" .
```

- Tên tham số.

```
:Tên_tham_số rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Tham_số ;  
    rdfs:range xsd:string ;  
    rdfs:label "Tên tham số" .
```

- Tên hàm

```
:Tên_hàm rdf:type owl:DatatypeProperty ;
```

```
rdfs:domain :Hàm ;  
rdfs:range xsd:string ;  
rdfs:label "Tên hàm" .
```

- Kiểu trả về

```
:Kiểu_trả_về rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Hàm ;  
    rdfs:range xsd:string ;  
    rdfs:label "Kiểu trả về" .
```

- Tham số đầu vào.

```
:Tham_số_đầu_vào rdf:type owl:ObjectProperty ;  
    rdfs:domain :Hàm ;  
    rdfs:range :Tham_số ;  
    rdfs:label "Tham số đầu vào" .
```

- Điều kiện dừng

```
:Điều_kiện_dừng rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Cấu_trúc_Lặp ;  
    rdfs:range xsd:string ;  
    rdfs:label "Điều kiện dừng" .
```

- Số lần gọi đệ quy.

```
:Số_lần_gọi_đệ_quy rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Hàm ;  
    rdfs:range xsd:int ;  
    rdfs:label "Số lần gọi đệ quy" .
```

- Tên hàm thư viện

```
:Tên_hàm_thư_viện rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Hàm_thư_viện ;  
    rdfs:range xsd:string ;  
    rdfs:label "Tên hàm thư viện" .
```

- Mô tả chức năng

```
:Mô_tả_chức_năng rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Hàm ;
```

```
    rdfs:range xsd:string ;  
    rdfs:label "Mô tả chức năng" .
```

- Tên kiểu dữ liệu

```
:Tên_kiểu_dữ_liệu rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Kiểu_dữ_liệu ;  
    rdfs:range xsd:string ;  
    rdfs:label "Tên kiểu dữ liệu" .
```

- Kích thước

```
:Kích_thước rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Kiểu_dữ_liệu ;  
    rdfs:range xsd:int ;  
    rdfs:label "Kích thước" .
```

- Mô tả

```
:Mô_tả rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Cấu_trúc_Dữ_liệu ;  
    rdfs:range xsd:string ;  
    rdfs:label "Mô tả" .
```

- Cấu trúc dữ liệu.

```
:Cấu_trúc_dữ_liệu rdf:type owl:ObjectProperty ;  
    rdfs:domain :Cấu_trúc_Dữ_liệu ;  
    rdfs:range :Thành_phần_Dữ_liệu ;  
    rdfs:label "Cấu trúc dữ liệu" .
```

- Điều hướng con trỏ

```
:Điều_hướng_con_trỏ rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Con_trỏ ;  
    rdfs:range xsd:string ;  
    rdfs:label "Điều hướng con trỏ" .
```

- Kích thước mảng

```
:Kích_thước_mảng rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Mảng ;  
    rdfs:range xsd:int ;
```

rdfs:label "Kích thước mảng" .

- Số lượng phần tử.

:Số_lượng_phần_tử rdf:type owl:DatatypeProperty ;
rdfs:domain :Cấu_trúc_Danh_sách ;
rdfs:range xsd:int ;
rdfs:label "Số lượng phần tử" .

- Loại lệnh điều kiện

:Loại_lệnh_điều_kiện rdf:type owl:DatatypeProperty ;
rdfs:domain :Lệnh_điều_kiện ;
rdfs:range xsd:string ;
rdfs:label "Loại lệnh điều kiện" .

- Điều kiện kiểm tra.

:Điều_kiện_kiểm_tra rdf:type owl:DatatypeProperty ;
rdfs:domain :Lệnh_điều_kiện ;
rdfs:range xsd:string ;
rdfs:label "Điều kiện kiểm tra" .

- Biểu thức điều kiện,

:Biểu_thức_điều_kiện rdf:type owl:DatatypeProperty ;
rdfs:domain :Lệnh_điều_kiện ;
rdfs:range xsd:string ;
rdfs:label "Biểu thức điều kiện" .

- Khối mã thực thi nếu điều kiện đúng.

:Khối_mã_thực_thi_nếu_điều_kiện_đúng rdf:type owl:DatatypeProperty ;
rdfs:domain :Lệnh_điều_kiện ;
rdfs:range xsd:string ;
rdfs:label "Khối mã thực thi nếu điều kiện đúng" .

- Khối mã thực thi nếu điều kiện "if" sai.

:Khối_mã_thực_thi_nếu_điều_kiện_sai rdf:type owl:DatatypeProperty ;
rdfs:domain :Lệnh_điều_kiện ;
rdfs:range xsd:string ;
rdfs:label "Khối mã thực thi nếu điều kiện 'if' sai" .

- Giá trị chuyển đổi

```
:Giá_trị_chuyển_đổi rdf:type owl:DatatypeProperty ;
    rdfs:domain :Biểu_thức_chuyển_đổi ;
    rdfs:range xsd:string ;
    rdfs:label "Giá trị chuyển đổi" .
```

- Các trường hợp và mã thực thi tương ứng.

```
:Các_trường_hợp_và_mã_thực_thi_tương_ứng rdf:type owl:DatatypeProperty ;
    rdfs:domain :Biểu_thức_Switch ;
    rdfs:range xsd:string ;
    rdfs:label "Các trường hợp và mã thực thi tương ứng" .
```

- Loại vòng lặp

```
:Loại_vòng_lặp rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Loại vòng lặp" .
```

- Điều kiện dừng

```
:Điều_kiện_dừng rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Điều kiện dừng" .
```

- Khởi lặp.

```
:Khởi_lặp rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Khởi lặp" .
```

- Biểu thức khởi đầu

```
:Biểu_thức_khởi_đầu rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Biểu thức khởi đầu" .
```

- Điều kiện lặp

```
:Điều_kiện_lặp rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Điều kiện lặp" .
```

- Bước nhảy và khối mã thực thi.

```
:Bước_nhảy_và_khối_mã_thực_thi rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Bước nhảy và khối mã thực thi" .
```

- Biểu thức điều kiện và khối mã thực thi.

```
:Bước_nhảy_và_khối_mã_thực_thi rdf:type owl:DatatypeProperty ;
    rdfs:domain :Cấu_trúc_Lặp ;
    rdfs:range xsd:string ;
    rdfs:label "Bước nhảy và khối mã thực thi" .
```

- Kiểu con trỏ

```
:Kiểu_con_trỏ rdf:type owl:DatatypeProperty ;
    rdfs:domain :Con_trỏ ;
    rdfs:range xsd:string ;
    rdfs:label "Kiểu con trỏ" .
```

- Địa chỉ trỏ tới.

```
:Địa_chỉ_trỏ_tới rdf:type owl:DatatypeProperty ;
    rdfs:domain :Con_trỏ ;
    rdfs:range xsd:string ;
    rdfs:label "Địa chỉ trỏ tới" .
```

- Kiểu dữ liệu con trỏ

```
:Kiểu_dữ_liệu_con_trỏ rdf:type owl:DatatypeProperty ;
    rdfs:domain :Con_trỏ ;
    rdfs:range xsd:string ;
    rdfs:label "Kiểu dữ liệu con trỏ" .
```

- Kiểu dữ liệu của mảng

```
:Kiểu_dữ_liệu_của_mảng rdf:type owl:DatatypeProperty ;
```



```
rdfs:domain :Mảng ;  
rdfs:range xsd:string ;  
rdfs:label "Kiểu dữ liệu của mảng" .
```

- Phần tử.

```
:Phần_tử rdfs:type owl:ObjectProperty ;  
rdfs:domain :Mảng ;  
rdfs:range :Biến ;  
rdfs:label "Phần tử" .
```

- Giá trị các phần tử.

```
:Giá_trị_các_phần_tử rdfs:type owl:DatatypeProperty ;  
rdfs:domain :Mảng ;  
rdfs:range xsd:string ;  
rdfs:label "Giá trị các phần tử" .
```

- Tên thư viện

```
:Tên_thư_viện rdfs:type owl:DatatypeProperty ;  
rdfs:domain :Thư_viện ;  
rdfs:range xsd:string ;  
rdfs:label "Tên thư viện" .
```

- Hàm có sẵn.

```
:Hàm_có_sẵn rdfs:type owl:DatatypeProperty ;  
rdfs:domain :Hàm_thư_viện ;  
rdfs:range xsd:boolean ;  
rdfs:label "Hàm có sẵn" .
```

- Loại cấu trúc dữ liệu

```
:Loại_cấu_trúc_dữ_liệu rdfs:type owl:DatatypeProperty ;  
rdfs:domain :Cấu_trúc_Dữ_liệu ;  
rdfs:range xsd:string ;  
rdfs:label "Loại cấu trúc dữ liệu" .
```

- Các thành phần dữ liệu.

```
:Các_thành_phần_dữ_liệu rdfs:type owl:ObjectProperty ;
```

rdfs:domain :Cấu_trúc_Dữ_liệu ;
rdfs:range :Thành_phần_Dữ_liệu ;
rdfs:label "Các thành phần dữ liệu" .

- Loại danh sách liên kết (singly linked list, doubly linked list)

:Loại_danh_sách_liên_kết rdfs:type owl:DatatypeProperty ;
rdfs:domain :Danh_sách_liên_kết ;
rdfs:range xsd:string ;
rdfs:label "Loại danh sách liên kết" .

- Các phần tử và liên kết giữa chúng.

:Các_phần_tử_và_liên_kết rdfs:type owl:DatatypeProperty ;
rdfs:domain :Danh_sách_liên_kết ;
rdfs:range xsd:string ;
rdfs:label "Các phần tử và liên kết giữa chúng" .

- Loại ngăn xếp

:Loại_ngăn_xếp rdfs:type owl:DatatypeProperty ;
rdfs:domain :Ngăn_xếp ;
rdfs:range xsd:string ;
rdfs:label "Loại ngăn xếp" .

- Các phần tử và quy tắc hoạt động.

:Các_phần_tử_và_quy_tắc_hoạt_động rdfs:type owl:DatatypeProperty ;
rdfs:domain :Ngăn_xếp ;
rdfs:range xsd:string ;
rdfs:label "Các phần tử và quy tắc hoạt động" .

- Loại hàng đợi (queue)

:Loại_hàng_đội rdfs:type owl:DatatypeProperty ;
rdfs:domain :Hàng_đội ;
rdfs:range xsd:string ;
rdfs:label "Loại hàng đợi" .

- Các phần tử và quy tắc hoạt động.

:Các_phần_tử_và_quy_tắc_hoạt_động rdfs:type owl:DatatypeProperty ;
rdfs:domain :Hàng_đội ;

```
rdfs:range xsd:string ;  
rdfs:label "Các phần tử và quy tắc hoạt động" .
```

- Nội dung ghi chú

```
:Nội_dung_ghi_chú rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Ghi_chú ;  
    rdfs:range xsd:string ;  
    rdfs:label "Nội dung ghi chú" .
```

- Vị trí.

```
:Vị_trí rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Ghi_chú ;  
    rdfs:range xsd:string ;  
    rdfs:label "Vị trí" .
```

- Nội dung công việc cần thực hiện.

```
:Nội_dung_công_việc_cần_thực_hiện rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Công_việc ;  
    rdfs:range xsd:string ;  
    rdfs:label "Nội dung công việc cần thực hiện" .
```

- Nội dung thông tin gỡ lỗi.

```
:Nội_dung_thông_tin_gỡ_lỗi rdf:type owl:DatatypeProperty ;  
    rdfs:domain :Thông_tin_gỡ_lỗi ;  
    rdfs:range xsd:string ;  
    rdfs:label "Nội dung thông tin gỡ lỗi" .
```

1.3 Đối tượng (Individual)

Các đối tượng biểu diễn các thực thể cụ thể hoặc ví dụ của các khái niệm trong lập trình C

- Biến (Variable):

```
:biến1 rdf:type :Biến ;  
    :Tên "x" ;  
    :Kiểu "int" ;  
    :Giá_trị "5" .
```

```
:biến2 rdf:type :Biến ;
```

:Tên "y" ;
:Kiểu "double" ;
:Giá_trị "3.14" .

- Hàm (Function)

:hàm1 rdf:type :Hàm ;
:Tên "tính_tổng" ;
:Kiểu_trả_về "int" .

:hàm2 rdf:type :Hàm ;
:Tên "in_ra" ;
:Kiểu_trả_về "void" .

- Câu lệnh (Statements)

:câu_lệnh1 rdf:type :Câu_lệnh ;
:Nội_dung "if (x > 0) { ... }" .

:câu_lệnh2 rdf:type :Câu_lệnh ;
:Nội_dung "for (i = 0; i < n; i++) { ... }" .

- Kiểu dữ liệu (Data type)

:kiểu_int rdf:type :Kiểu_dữ_liệu ;
:Tên "int" .

:kiểu_double rdf:type :Kiểu_dữ_liệu ;
:Tên "double" .

- Mảng (Arrays)

:mảng1 rdf:type :Mảng ;
:Tên "arr1" ;
:Kiểu "int[]" ;
:Kích_thước 10 .

:mảng2 rdf:type :Mảng ;
:Tên "arr2" ;
:Kiểu "char[]" ;
:Kích_thước 20 .

- Chuỗi (Strings)

:chuỗi1 rdf:type :Chuỗi ;
:Tên "str1" ;

:Giá_trị "Hello, World!" .

:chuỗi2 rdf:type :Chuỗi ;
:Tên "str2" ;
:Giá_trị "Programming is fun." .

- Con trỏ (Pointers)

:con_trỏ1 rdf:type :Con_trỏ ;
:Tên "ptr1" ;
:Kiểu "int*" .

:con_trỏ2 rdf:type :Con_trỏ ;
:Tên "ptr2" ;
:Kiểu "char*" .

- Cấu trúc dữ liệu (Data Structures)

:cấu_trúc1 rdf:type :Cấu_trúc_Danh_sách ;
:Tên "list1" ;
:Số_lượng_phần_tử 100 .

:cấu_trúc2 rdf:type :Cấu_trúc_Danh_sách ;
:Tên "list2" ;
:Số_lượng_phần_tử 50 .

- Lệnh điều kiện (Conditional Statements)

:lệnh_điều_kiện1 rdf:type :Lệnh_điều_kiện ;
:Nội_dung "if (x > 0) { ... }" .

:lệnh_điều_kiện2 rdf:type :Lệnh_điều_kiện ;
:Nội_dung "if (y == 0) { ... }" .

- Cấu trúc lặp (Loop Structures)

:cấu_trúc_lặp1 rdf:type :Cấu_trúc_Lặp ;
:Nội_dung "for (i = 0; i < n; i++) { ... }" .

:cấu_trúc_lặp2 rdf:type :Cấu_trúc_Lặp ;
:Nội_dung "while (x > 0) { ... }" .

1.4 Quan hệ (Relations)

Quan hệ xác định mối quan hệ giữa lớp và đối tượng

- Kế thừa (Inheritance): Sử dụng biểu thị mối quan hệ giữa một biến và nơi nó được sử dụng trong mã nguồn.

:Lớp_con rdf:type owl:Class .

:Lớp_cha rdf:type owl:Class .

:Kế_thừa rdf:type owl:ObjectProperty ;

 rdfs:domain :Lớp_con ;

 rdfs:range :Lớp_cha ;

 rdfs:label "Kế thừa từ" .

- Kết nối (Linkage): Kết nối biểu thị mối quan hệ giữa các tệp và thư viện trong lập trình C.

:Tệp rdf:type owl:Class .

:Thư_viện rdf:type owl:Class .

:Kết_nối rdf:type owl:ObjectProperty ;

 rdfs:domain :Tệp ;

 rdfs:range :Thư_viện ;

 rdfs:label "Kết nối" .

- Có thành phần của (Composition): Quan hệ này biểu diễn mối quan hệ giữa một đối tượng lớn hơn và các thành phần con của nó. Ví dụ, một cấu trúc (struct) trong lập trình C có thể chứa các biến khác.

:Đối_tượng_Lớn rdf:type owl:Class .

:Đối_tượng_Nhỏ rdf:type owl:Class .

:Có_thành_phần_của rdf:type owl:ObjectProperty ;

 rdfs:domain :Đối_tượng_Lớn ;

 rdfs:range :Đối_tượng_Nhỏ ;

 rdfs:label "Có thành phần của" .

- Là thành phần của (Part-Of): Quan hệ này là ngược của "Có Thành Phần Của," biểu diễn mối quan hệ giữa một thành phần và đối tượng lớn hơn mà nó thuộc về.

:Đối_tượng_Lớn rdf:type owl:Class .

:Đối_tượng_Nhỏ rdf:type owl:Class .

:Là_thành_phần_của rdf:type owl:ObjectProperty ;

 rdfs:domain :Đối_tượng_Nhỏ ;

 rdfs:range :Đối_tượng_Lớn ;

 rdfs:label "Là thành phần của" .

- Gọi hàm (Function Call): Quan hệ này biểu diễn khi một hàm được gọi từ một điểm trong mã nguồn.

:Hàm rdf:type owl:Class .

:Câu_lệnh rdf:type owl:Class .

:Gọi_hàm rdf:type owl:ObjectProperty ;

rdfs:domain :Câu_lệnh ;

rdfs:range :Hàm ;

rdfs:label "Gọi hàm" .

- Tham số của (Parameter Of): Quan hệ này biểu diễn khi một biến hoặc tham số được truyền vào một hàm hoặc phương thức.

:Hàm rdf:type owl:Class .

:Biến rdf:type owl:Class .

:Tham_số_của rdf:type owl:ObjectProperty ;

rdfs:domain :Biến ;

rdfs:range :Hàm ;

rdfs:label "Tham số của" .

- Thuộc về (Belongs to): Quan hệ này biểu diễn mối quan hệ giữa một đối tượng và đối tượng lớn hơn mà nó thuộc về, ví dụ như một biến thuộc về một cấu trúc.

:Đối_tượng rdf:type owl:Class .

:Đối_tượng_Lớn rdf:type owl:Class .

:Thuộc_về rdf:type owl:ObjectProperty ;

rdfs:domain :Đối_tượng ;

rdfs:range :Đối_tượng_Lớn ;

rdfs:label "Thuộc về" .

- Gán giá trị cho (Assigns Value to): Quan hệ này biểu diễn khi một giá trị được gán cho một biến.

:Biến rdf:type owl:Class .

:Gán_giá_trị_cho rdf:type owl:ObjectProperty ;

rdfs:domain :Biến ;

rdfs:range xsd:string ; # Hoặc kiểu dữ liệu tương ứng

rdfs:label "Gán giá trị cho" .

- Là kiểu dữ liệu của (Is Data Type of): Quan hệ này biểu diễn khi một đối tượng được định nghĩa là một kiểu dữ liệu cụ thể.

:Đối_tượng rdf:type owl:Class .

:Kiểu_dữ_liệu rdf:type owl:Class .

:Là_kiểu_dữ_liệu_của rdf:type owl:ObjectProperty ;

 rdfs:domain :Đối_tượng ;

 rdfs:range :Kiểu_dữ_liệu ;

 rdfs:label "Là kiểu dữ liệu của" .

- Là kiểu trả về của (Is Return Type of): Quan hệ này biểu diễn khi một hàm hoặc phương thức được định nghĩa với kiểu dữ liệu trả về cụ thể.

:Hàm rdf:type owl:Class .

:Kiểu_dữ_liệu rdf:type owl:Class .

:Là_kiểu_trả_về_của rdf:type owl:ObjectProperty ;

 rdfs:domain :Hàm ;

 rdfs:range :Kiểu_dữ_liệu ;

 rdfs:label "Là kiểu trả về của" .

2. Giới thiệu ứng dụng Protégé

Protege là một ứng dụng phát triển và quản lý ontology mạnh mẽ và phổ biến. Protege là một công cụ quan trọng trong lĩnh vực nghiên cứu về ontologies và phát triển các ứng dụng Semantic Web. Nó được sử dụng rộng rãi trong cộng đồng khoa học máy tính và ngành công nghiệp để tạo ra và quản lý thông tin có ý nghĩa về ngữ nghĩa.

Mục tiêu chính: Protege được phát triển để hỗ trợ người dùng xây dựng, quản lý, và thực hiện ontology. Ontology là một phần quan trọng của thế giới Semantic Web và được sử dụng rộng rãi trong các ứng dụng như khai thác dữ liệu, truy vấn thông tin, và trí tuệ nhân tạo.

Giao diện người dùng đồ họa: Protege cung cấp một giao diện người dùng đồ họa dễ sử dụng, cho phép người dùng tạo và chỉnh sửa ontology một cách trực quan. Điều này giúp giảm sự phức tạp trong việc tạo và quản lý ontology.

Hỗ trợ nhiều ngôn ngữ và định dạng: Protege hỗ trợ nhiều ngôn ngữ ontology quan trọng như OWL (Web Ontology Language) và RDF (Resource Description Framework). Điều này cho phép bạn xây dựng ontology phức tạp và phong phú.

Tích hợp mở rộng: Protege có một cộng đồng phát triển mở rộng và plugin đáng kể. Bạn có thể tìm thấy một loạt các plugin để mở rộng chức năng và tính năng của Protege theo nhu cầu cụ thể của bạn.

Hỗ trợ ghi chú và tài liệu: Protege cho phép bạn gắn chú thích và tài liệu cho các phần tử ontology, giúp giải thích và tài liệu hóa ontology của bạn cho người sử dụng khác.

Kiểm tra tính nhất quán: Protege có tích hợp nhiều công cụ và tiện ích giúp bạn kiểm tra tính nhất quán và chất lượng của ontology.

Cộng đồng và tài liệu: Protege có một cộng đồng sử dụng lớn và tài liệu phong phú. Bạn có thể tìm thấy tài liệu hướng dẫn, hướng dẫn sử dụng, và sự hỗ trợ từ cộng đồng khi bạn gặp vấn đề.

Công cụ trực tuyến và API: Ngoài ứng dụng máy tính để bàn, Protege cũng cung cấp một số dịch vụ trực tuyến và API để tích hợp ontology vào các ứng dụng web và dự án khác.

III. Kết luận

Ứng dụng tổ chức tri thức lập trình C bằng ontology đã hoàn thành và đã đem lại lợi ích lớn cho người dùng trong việc tìm kiếm thông tin và học hỏi về lập trình C. Ontology giúp tổ chức tri thức một cách cấu trúc và dễ quản lý, cung cấp một nền tảng mạnh mẽ cho ứng dụng này. Chúng tôi sẽ tiếp tục phát triển và cải tiến ứng dụng để đáp ứng các nhu cầu ngày càng đa dạng của cộng đồng lập trình.

IV. Tài liệu tham khảo

- Biểu Diễn Tri Thức Và Suy Luận (NXB Đại Học Quốc Gia 2016) - Đỗ Văn Nhơn
Tìm hiểu và nghiên cứu về ontology và ứng dụng trong công nghệ tri thức
- "Programming in C" bởi Stephen G. Kochan.
- W3C OWL Web Ontology Language Guide (<https://www.w3.org/TR/owl-guide/>)
- Protege (<https://protege.stanford.edu/>)