**CẤU TRÚC ĐIỀU KHIỂN TRONG NGÔN NGỮ LẬP TRÌNH**

**Thành viên thực hiện:**

20521658 – Trương Đăng Nghĩa

**Nội dung:**

* Giới thiệu.
* Các câu lệnh lựa chọn:
  + Two-way selectors.
  + Multiple-way selectors.
  + Nesting selectors.
* Các câu lệnh lặp:
  + Pre-test: while, for loop.
  + Post-test: do-while loop.
* Rẽ nhánh không điều kiện: goto, break, continue.
* Cấu trúc gọi đệ quy.
* Kết luận.

**Giới thiệu:**

Dữ liệu và tác vụ là hai yếu tố cơ bản của chương trình. Mỗi sự kết hợp của chúng gắn liền với cấu trúc điều khiển. Cấu trúc điều khiển là tập hợp các qui tắc xác định thứ tự thực hiện chương trình.

–  Xét về cấu trúc, có 3 loại điều khiển:

+ Điều khiển trong các biểu thức.  
+ Điều khiển giữa các phát biểu: cấu trúc điều kiện/ cấu trúc lặp.  
+ Điều khiển trong chương trình con: gọi trả về hay đệ qui.

– Xét về thiết kế ngôn ngữ, có 2 loại điều khiển:

+ Điều khiển ngầm: được thiết kế trong ngôn ngữ lập trình.  
(Ví dụ: quy tắc ưu tiên các toán tử)

+ Điều khiển tường minh: được xác định bởi lập trình viên.  
(Ví dụ: phát biểu GOTO, CONTINUE, BEAK,…)

Các vấn đề logic được giải quyết bằng cách phối hợp 3 cấu trúc:

– Cấu trúc tuần tự (Sequence structures).  
– **Cấu trúc chọn** (Selection structures).  
– **Cấu trúc lặp** (Iterative structures).

Diagram

Description automatically generated

– Lệnh (statement): một biểu thức kết thúc bởi 1 dấu chấm phẩy gọi là 1 lệnh.

vd:

 Graphical user interface, text, application

Description automatically generated

***C++*** ***Python***

– Khối lệnh (block): một hay nhiều lệnh được bao quanh bởi cặp dấu { } gọi là một khối lệnh. Về mặt cú pháp, khối lệnh tương đương 1 câu lệnh đơn.

Graphical user interface, text, application

Description automatically generated Text

Description automatically generated

***C++*** ***Python***

Các câu lệnh lựa chọn

Một câu lệnh lựa chọn cung cấp các lựa chọn giữa hai hay nhiều cách thực thi trong một chương trình. Lệnh lựa chọn (lệnh điều khiển) là một lệnh biểu thị sự lựa chọn của hai hoặc đa nhánh để thực hiện.

Chia làm 2 loại:

+ Chỉ có 2 lựa chọn (lệnh **if**).  
+ Nhiều lựa chọn (lệnh **case**).

“Các câu lệnh lựa chọn” là cầu trúc được trình bày đầu tiên trong cấu trúc điều khiển trình tự giữa các phát biểu, sau đó chúng ta sẽ tìm hiểu và phân tích “các câu lệnh lặp” và phần cuối cùng là “rẽ nhánh không điều kiện”.

**1. Cấu trúc lệnh điều khiển**:

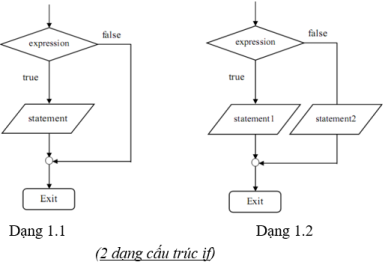
*IF … ELSE – (Two-Way Selection Statements)*

– Dạng phổ biến:

**if** control\_expression  **then** statement  
**else** statement

– Yếu tố thiết kế:

+ **Dạng và kiểu của biểu thức** lựa chọn như thế nào, quan hệ, toán và logic?  
+ Các câu lệnh theo sau **then** và **else**?  
+ **Lệnh lựa chọn có lồng nhau hay không**?



*Dạng và kiểu của biểu thức*

– Cấu trúc **if**ở dạng 1.1 cho thấy nếu biểu thức điều kiện có giá trị “true” thì statement được thực hiện, ngược lại không làm gì cả. Tại dạng 1.2 nếu biểu thức có giá trị “true” thì statement 1 được thực hiện, ngược lại statement 2 được thực hiện.

– Java, C#, C (89 trở về trước): Chỉ dùng biểu thức logic.  
– C (99), C++ và Python: biểu thức toán và logic

Text

Description automatically generated

Kq: result = 5

**2. Nesting selectors:**

– Ví dụ 1b, đoạn chương trình C++:

Text

Description automatically generated

Kq: result = 1

Ngôn ngữ C, C++, C#, Java quy định lệnh **if** và lệnh **else** gần nhau nhất sẽ đi cùng nhau.

Ví dụ 2b:

Text

Description automatically generated Text

Description automatically generated with medium confidence

Kq: result = 5 Kq: result = 1

Ngôn ngữ Python quy định lệnh if và lệnh else cùng hàng sẽ đi cùng nhau.

Diagram

Description automatically generated with low confidence  
Vì lệnh else sẽ đi cùng với lệnh if gần nhất, do đó để giảm và tránh nhầm lẫn ta sẽ viết thẳng hàng if(a>b) với else hoặc có thể đặt if(a>b) x=a vào trong khối lệnh {}.

Cấu trúc if lồng nhau càng nhiều, cấp độ phức tạp càng cao, chương trình chạy càng chậm và trong lúc lập trình dễ bị nhầm lẫn.

**3. Lệnh lựa chọn đa nhánh** – (Multiple-Selection Statements)

Cho phép lựa chọn 1 nhánh trong số nhiều nhánh lệnh để thực hiện.  
Các yếu tố trong lệnh lựa chọn đa nhánh:

+ Kiểu và dạng của biểu thức điều khiển?  
+ Công việc của từng nhánh lệnh?  
+ Có nhánh lệnh nào không thỏa điều kiện? Nếu có nó thực hiện lệnh gì?

**a) Mô hình switch-case**

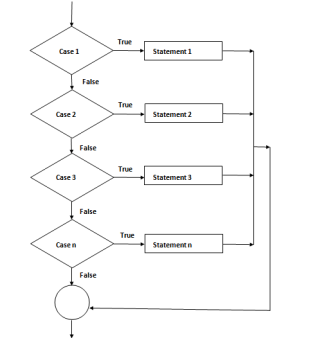
**switch** (expression){  
**case** constant\_expression1 : statement1; break;  
…  
**case** constant\_expressionn : statement\_n; break;  
[**default**: statementn+1]  
}

Tương ứng với từng giá trị của expression mà statementi được thực hiện. Trong trường hợp giá trị biểu thức không trùng với giá trị nào thì statementn+1được thực hiện.

Chú ý:

+ expression trong switch( ) phải có kiểu dữ liệu đếm được như char, int,.. không thể là kiểu thực (double, float).

+ Sau mỗi statement phải có lệnh break, nếu không thì các statement lần lượt được thực thi cho đến khi gặp 1 lệnh break khác.



– Switch-case trong C, C++, Java:

+ Biểu thức điều khiển phải là kiểu nguyên.  
+ Câu lệnh lựa chọn có thể là 1 câu lệnh đơn hoặc khối lệnh.  
+ Kết thúc cấu trúc switch-case khi gặp break;  
+ Default: được chọn nếu không có giá trị nào thỏa expression.

Ví dụ:

A picture containing chart

Description automatically generated

***C++***

Text

Description automatically generated

***Python***

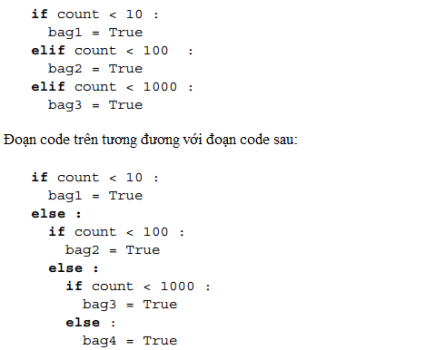
**b) Mô hình if**

Trong nhiều trường hợp, khi sử dụng lệnh switch-case sẽ không phù hợp với đa lựa chọn (ngoại trừ Ruby). Một ví dụ, khi thực hiện việc chọn lựa dựa trên một biểu thức Boolean (đúng hoặc sai) ta sử dụng cấu trúc if else lồng nhau để mô phỏng như một lệnh lựa chọn đa nhánh (case).

Lệnh lựa chọn đa nhánh có thể chuyển thành lệnh chọn 2 nhánh với else-if

if (expr == const\_expr\_1) stmt\_1;  
else if (expr == const\_expr\_2) stmt\_2;  
…  
else if (expr == const\_expr\_n) stmt\_n;  
else stmt\_n+1;

Ví dụ trong Python (trong Python else-if được viết thành elif ):

[](https://ongthovuive.files.wordpress.com/2016/12/cautrucdieukhien13.png)

**Các câu lệnh lặp**

**Graphical user interface

Description automatically generated with medium confidence**

Đôi lúc chúng ta viết một đoạn code lặp đi lặp lại nhiều lần để thực hiện một yêu cầu nào đó, thật vất vả khi bạn phải viết 5, 10 hay thậm chí 1000 câu lệnh giống nhau chỉ để làm một công việc. Chính vì vậy, khái niệm lệnh lặp giúp rất nhiều cho lập trình viên để xây dựng các chương trình khác nhau một cách hiệu quả, nhanh chóng.

Ở phần này, chúng ta sẽ chú ý đến ba cấu trúc: for, while, do-while. Ba cấu trúc này đóng góp rất lớn trong quá trình cài đặt phần mềm, loại bỏ nhanh chóng việc giảm hiệu xuất gây ra do sử dụng quá nhiều dòng lệnh có chung tính chất.

Diagram

Description automatically generated

– Điều kiện kết thúc sự lặp có thể được xác định thông qua biến đếm hoặc là điều kiện phức tạp được biểu diễn bằng một biểu thức luận lý. Có 2 loại lệnh lặp: Lệnh lặp với bộ đếm và lệnh lặp có điều kiện.

– Các yếu tố trong lệnh lặp

+ Làm thế nào để kiểm soát lặp?  
+ Kiểm soát lặp xuất hiện ở đâu trong vòng lặp?

Lệnh lặp với bộ đếm – (Counter-Controlled Loops)

Trong lệnh điều khiển lặp có một biến số, ta gọi đây là biến lặp. Trong đó giá trị của biến đếm từ giá trị bắt đầu (initial) đến giá trị kết thúc (terminal) và giá trị của bước nhảy (stepsize).

– Các yếu tố trong lệnh lặp:

+ Kiểu của biến lặp và phạm vi?  
+ Giá trị của biến lặp khi vòng lặp kết thúc?  
+ Giá trị của biến lặp có được thay đổi trong thân vòng lặp không? Nếu có thì có ảnh hưởng đến vòng lặp không?

– Cú pháp lệnh for trong C:

for ([expr\_1] ; [expr\_2]; [expr\_3])  
statement

+ Mọi thứ có thể thay đổi trong thân vòng lặp.  
+ Biểu thức expr\_1 được định lượng 1 lần trước khi tực hiện vòng lặp ( biểu thức khởi tạo).

+ Sau mỗi lần lặp, expr\_2 (biểu thức điều kiện) và expr\_3 (biểu thức điều khiển lặp) được định lượng.  
+ Xóa các biểu thức expr\_1, expr\_2, expr\_3 trong vòng for sẽ cho vòng lặp vô tận.  
+ C++ cho phép khai báo kiểu trong expr\_1.

for (int count=0;count<len; count++)   {…}

Logo, company name

Description automatically generated

**C++**

Text

Description automatically generated

**Python**

Lệnh lặp có điều kiện – (Logically Controlled Loops)

– Chỉ được thực hiện khi điều kiện đúng  
– Các yếu tố trong lệnh lặp có điều kiện:

+ Kiểm tra điều kiện trước hay sau.  
+ Là trường hợp đặc biệt của lệnh lặp với bộ đếm?

– Cú pháp

Pre-test Loops:

While (ctrl\_expr)

loop body

và

Post-test Loops:

Do {

loop body

} while (ctrl\_expr);

– C và C++ dùng cả “while-do” – (lặp nếu điều kiện còn đúng) và “do-while” – ( lặp đến khi điều kiện sai).

==> Nhận định qua 2 loại lệnh lặp: Các phát biểu lặp vẫn được hiện thực bằng các lệnh nhảy có điều kiện và không điều kiện của ngôn ngữ máy.

**Rẽ nhánh không điều kiện**

– Lệnh rẽ nhánh không điều kiện được đưa ra vào những năm cuối thập niên 60.  
– Cho phép thay đổi thứ tự thực hiện chương trình.  
– Cơ chế phổ biến nhất là lệnh: goto  
– Một số NNLT không hỗ trợ lệnh goto  
– Lệnh goto làm chương trình khó đọc và khó bảo trì.

– Cho phép thoát khỏi luồng điều khiển thông thường

***Lệnh continue:***

– Lệnh continue sẽ kết thúc vòng lặp hiện hành và chuyển đến vòng lặp tiếp theo.  
– Khi gặp lệnh này, các câu lệnh còn lại trong thân của vòng lặp sẽ được bỏ qua.  
Vd:  
A picture containing schematic

Description automatically generated

**C++**

Text

Description automatically generated

**Python**

***Lệnh break:***

– Lệnh break sẽ thoát khỏi cấu trúc lặp hoặc switch

Text

Description automatically generated

**C++**

Text

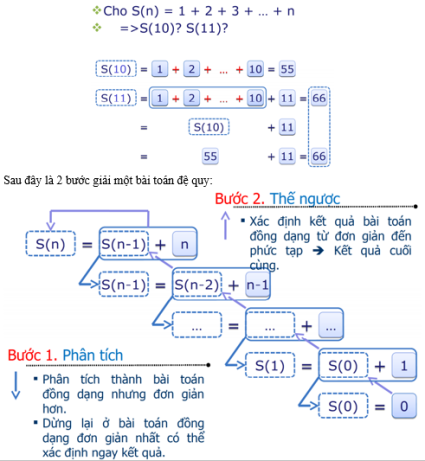
Description automatically generated

**Python**

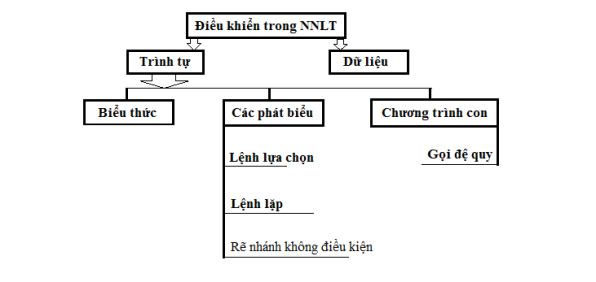
**Cấu trúc gọi đệ quy**

* Là cấu trúc điều khiển trình tự quan trọng trong lập trình.
* Gọi đến nhiều bản hoạt động của cùng một chương trình con có thể cùng tồn tại trong bộ nhớ, trong thời gian thực hiện chương trình.
* Nhiều giải thuật diễn đạt bằng đệ quy, vừa đơn giản, giảm chi phí lập trình hơn so với các giải quyết bằng sự lặp.
* Một chương trình con (hay hàm) được gọi là đệ quy nếu bên trong thân nó có một lời gọi đến chính nó và khi đạt được tại điều kiện dừng, quá trình đệ quy sẽ không diễn ra nữa.

Xét ví dụ:

[](https://ongthovuive.files.wordpress.com/2016/12/cautrucdieukhien14.png)

**Kết Luận**



Trong cấu trúc điều khiển trong ngôn ngữ lập trình bao gồm trình tự và dữ liệu.

Về mặt trình tự có thể phân ra làm 3 phần:

-Biểu thức

-Các phát biểu:

+Lệnh lựa chọn

+Lệnh lặp

+Rẽ nhánh không điều kiện

-Chương trình con: cấu trúc gọi đệ quy

***Tài liệu tham khảo:***

* **Các cấu trúc điều khiển trong ngôn ngữ lập trình**

https://ongthovuive.wordpress.com/2016/12/05/cac-cau-truc-dieu-khien-trong-ngon-ngu-lap-trinh/

* **Statement-Level Control Structures**

https://courses.uit.edu.vn/pluginfile.php/330259/mod\_resource/content/1/Ch08-jhchen%20Chapter%208%20Statement-Level%20Control%20Structures.pdf