

» CÁC CẤU TRÚC ĐIỀU KHIỂN

Name: Nguyễn Chí Hiếu

Date: 2020

» NỘI DUNG

1. Các khái niệm cơ bản

2. Cấu trúc tuần tự

3. CẤU TRÚC Rẽ NHÁNH

4. CẤU TRÚC LẶP

» CÁC KHÁI NIỆM CƠ BẢN

Lệnh (*statement*)

Là một biểu thức hay câu lệnh gán và có dấu chấm phẩy cuối lệnh.

```
1  int  a = 5;  
2  int  b = 10;  
3  int  c = a + b;  
4  
5  Console.WriteLine("c = a + b = {0}", c);
```

» CÁC KHÁI NIỆM CƠ BẢN

Khối lệnh (*block*)

Đây là một tập hợp các lệnh được đặt trong cặp dấu ngoặc nhọn { ... }

- * Một khối lệnh có thể có hay không có các lệnh.
- * Khối lệnh có thể lồng nhau nhiều cấp.

```
1 {  
2     Khoi_lenh_1;  
3     {  
4         Khoi_lenh_2;  
5     }  
6 }
```

» CÁC KHÁI NIỆM CƠ BẢN

Khối lệnh (*block*)

Chương trình có nhiều khối lệnh lồng nhau, các biến sẽ được truy xuất theo quy tắc sau:

```
1 {  
2     Khoi_lenh_1;  
3     {  
4         Khoi_lenh_2;  
5     }  
6 }
```

- * Khối lệnh 2 có thể sử dụng biến khai báo (*biến toàn cục*) trong khối lệnh 1.
- * Khối lệnh 1 *không* thể sử dụng biến khai báo (*biến cục bộ*) trong khối lệnh 2.

» CÁC KHÁI NIỆM CƠ BẢN

Khối lệnh (*block*)

```
1  {
2      int a, b;
3      a = 5;
4      {
5          a = 10;
6          b = 10;
7          int c = a + b;
8      }
9      Console.WriteLine("a = {0}", a); // 10
10     Console.WriteLine("b = {0}", b); // 10
11     Console.WriteLine("c = a + b = {0}", c); // Lỗi
12 }
```

» CẤU TRÚC TUẦN TỰ

Khái niệm

Đây là cấu trúc mặc định của một chương trình. Các thao tác được thực hiện tuần tự.

```
1  int a = int.Parse(Console.ReadLine());  
2  int b = int.Parse(Console.ReadLine());  
3  int c = a + b;  
4  
5  Console.Write("c = a + b = {0}", c);
```

```
> 5
```

```
> 10
```

```
> c = a + b = 15
```

» CẤU TRÚC RỄ NHÁNH (*branching*)

Khái niệm

Các thao tác thực hiện có thể có một hay nhiều nhánh khác nhau.

Phân loại cấu trúc rẽ nhánh

- * Cấu trúc rẽ nhánh có điều kiện: các thao tác được thực hiện hay không phụ thuộc vào kết quả của điều kiện đúng hay sai.
 - * if đơn, if ... else ..., if ... else if ... else ..., if lồng nhau
 - * switch ... case ...
- * Cấu trúc rẽ nhánh không điều kiện (*cấu trúc nhảy*):
 - * break: thoát khỏi vòng lặp for, while
 - * continue: thoát khỏi vòng lặp for, while

» CẤU TRÚC RỄ NHÁNH

Cấu trúc if

```
1  if (Dieu_kien)
2  {
3      Khoi_lenh;
4  }
```

- * Nếu điều kiện đúng, thực hiện khối lệnh.
- * Ngược lại, không thực hiện khối lệnh.

» CẤU TRÚC RỄ NHÁNH

```
1  int n = 27;  
2  if (n >= 0)  
3      Console.WriteLine("n la so nguyen duong");
```

```
> n la so nguyen duong
```

» CẤU TRÚC RỄ NHÁNH

Cấu trúc if

```
1  if (Dieu_kien)
2  {
3      Khoi_lenh_1;
4  }
5  else
6  {
7      Khoi_lenh_2;
8  }
```

* Nếu điều kiện đúng, thực hiện khối lệnh 1.

* Ngược lại, thực hiện khối lệnh 2.

» CẤU TRÚC RỄ NHÁNH

```
1  int n = 27;
2  if (n % 2 == 0)
3  {
4      Console.Write(n, "{0} la so chan");
5  }
6  else
7  {
8      Console.Write(n, "{0} la so le");
9  }
```

```
> 2  la so chan
```

» CẤU TRÚC RỄ NHÁNH

Cấu trúc if

```
1  if (Dieu_kien_1)
2  {
3      Khoi_lenh_1;
4  }
5  else if (
        Dieu_kien_2)
6  {
7      Khoi_lenh_2;
8  }
9  else
10 {
11     Khoi_lenh_3;
12 }
```

- * Nếu điều kiện 1 đúng, thực hiện khối lệnh 1.
- * Ngược lại, nếu điều kiện 2 đúng thực hiện khối lệnh 2.
- * Ngược lại, thực hiện khối lệnh 3.

» CẤU TRÚC RỄ NHÁNH

```
1  int n = 27;
2  if (n < 5)
3  {
4      Console.WriteLine("n < 5");
5  }
6  else if (n > 5)
7  {
8      Console.WriteLine("n > 5");
9  }
10 else
11 {
12     Console.WriteLine("n == 5");
13 }
```

> n > 5

» CẤU TRÚC RỄ NHÁNH

Cấu trúc if

```
1  if (Dieu_kien_1)
2  {
3      if (Dieu_kien_2)
4      {
5          Khoi_lenh;
6      }
7  }
```

- * Nếu điều kiện đúng, thực hiện khối lệnh.
- * Ngược lại, không thực hiện khối lệnh.

» CẤU TRÚC RỄ NHÁNH

```
1  int n = 27;
2  if (n < 5)
3  {
4      Console.WriteLine("n < 5");
5  }
6  else
7  {
8      if (n > 5)
9          Console.WriteLine("n > 5");
10     else
11         Console.WriteLine("n == 5");
12 }
```

> n > 5

» CẤU TRÚC RỄ NHÁNH

Cấu trúc switch ... case

```
1  switch (Ten_bien)
2  {
3      case Gia_tri_1:
4          Khoi_lenh_1;
5          break;
6      case Gia_tri_2:
7          Khoi_lenh_2;
8          break;
9      //...
10     default:
11         Khoi_lenh_n;
12         break;
13 }
```

- * Kiểm tra giá trị của biến tương ứng với từng trường hợp khác nhau.
- * Các trường hợp được kiểm tra theo thứ tự từ trên xuống.
- * Lệnh switch ... case sẽ dừng khi gặp lệnh break hay return

» CẤU TRÚC RỄ NHÁNH

```
1  int m = int.Parse(Console.ReadLine());
2  switch (m)
3  {
4      case 1:
5          Console.Write("Thang 1");
6          break;
7      case 2:
8          Console.Write("Thang 2");
9          break;
10     //...
11     default:
12         Console.Write("Thang không hợp lệ");
13         break;
14 }
```

» CẤU TRÚC RỄ NHÁNH

1. Giải phương trình bậc nhất $ax + b = 0$.
2. Nhập một ký tự từ bàn phím, kiểm tra ký tự này có phải ký tự số hay không?

» CẤU TRÚC LẶP (*looping*)

Khái niệm

Các thao tác sẽ được thực hiện nếu thỏa điều kiện lặp. Ngược lại, dừng quá trình lặp.

Phân loại cấu trúc lặp

- * Cấu trúc for
- * Cấu trúc while
- * Cấu trúc do ... while
- * Cấu trúc foreach

» CẤU TRÚC LẶP

Cấu trúc for

```
1  for (Khoi_tao_bien; Dieu_kien; Cap_nhat_bien)
2  {
3      Khoi_lenh;
4  }
```

- * Nếu điều kiện đúng, thực hiện khối lệnh.
- * Ngược lại, thoát khỏi vòng lặp.

» CẤU TRÚC LẶP

Cấu trúc for

```
1  int n = 10;
2  int s = 0;
3
4  for (int i = 1; i <= 10; i++)
5  {
6      s += i;
7  }
8  Console.WriteLine("sum = {0}", s);
```

```
> sum = 55
```

» CẤU TRÚC LẶP

Cấu trúc while

Kiểm tra điều kiện trước khi thực hiện

```
1 while (Dieu_kien)
2 {
3     Khoi_lenh;
4 }
```

- * Nếu điều kiện đúng, thực hiện khối lệnh.
- * Ngược lại, thoát khỏi vòng lặp.

» CẤU TRÚC LẶP

Cấu trúc while

```
1  int n = 10;
2  int s = 0;
3  int i = 1;
4  while (i <= n)
5  {
6      s += i;
7      i++;
8  }
9  Console.WriteLine("sum = {0}", s);
```

```
> sum = 55
```


» CẤU TRÚC LẶP

Cấu trúc do ... while

- * Kiểm tra điều kiện sau mỗi lần thực hiện.

```
1 do
2 {
3     Khoi_lenh;
4 } while (Dieu_kien);
```

- * Thực hiện khối lệnh.
- * Nếu điều kiện đúng, tiếp tục thực hiện vòng lặp. Ngược lại, thoát khỏi vòng lặp.

» CẤU TRÚC LẶP

Cấu trúc while

```
1  int n = 10;
2  int s = 0;
3  int i = 1;
4  do
5  {
6      s += i;
7      i++;
8  } while (i <= n);
9  Console.WriteLine("sum = {0}", s);
```

```
> sum = 55
```

» CẤU TRÚC LẶP

Cấu trúc foreach

```
1  foreach (kieu_du_lieu Ten_bien in tap_hop)
2  {
3      Khoi_lenh;
4  }
```

- * Thường sử dụng với các phần tử của một mảng, danh sách, tập hợp, ...
- * Khai báo biến tương ứng với kiểu dữ liệu của các phần tử trong tập hợp. Nếu chưa duyệt hết các phần tử thì tiếp tục thực hiện. Ngược lại, dừng vòng lặp.

» CẤU TRÚC LẶP

Cấu trúc foreach

```
1  string[] names = { "Tom", "Jerry" };
2
3  foreach (string name in names)
4  {
5      Console.WriteLine(name);
6  }
```

> Tom

> Jerry

» CẤU TRÚC LẶP

1. Tính $S(n) = 1 + 1/2 + 1/3 + \dots 1/n$
2. Tính n giai thừa.

» BÀI TẬP

1. Nhập vào một năm, cho biết năm đó thuộc thế kỷ thứ mấy?
2. Tìm số lớn nhất trong 3 số nguyên a , b và c .
3. Giải phương trình bậc hai $ax^2 + bx + c = 0, a \neq 0$.
4. Tính tiền điện phải trả khi biết số kWh. Giá bán lẻ điện được quy định như sau:
 - * 0-50: 1549 đồng/kWh
 - * 51-100: 1600
 - * 101-200: 1858
 - * 201-300: 2340
 - * 301-400: 2615
 - * 401 trở lên: 2701
5. Nhập vào một số nguyên dương n từ 0-99 và in ra số n bằng chữ. Ví dụ: $n = 27$ in ra màn hình dòng chữ "Hai mươi bảy".

» BÀI TẬP

6. In 256 ký tự của bảng mã ASCII.
7. Tính $S(n) = 1 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(n+2)}{6}$. Cho biết kết quả hai vế có bằng nhau không? Nhận xét.
8. Nhập vào một số nguyên dương n , tính tổng các ký tự số của n ?
9. Nhập vào một số nguyên dương n , kiểm tra n có phải là số nguyên tố hay không?
10. In bảng cửu chương ra màn hình.