

# Cơ sở lập trình

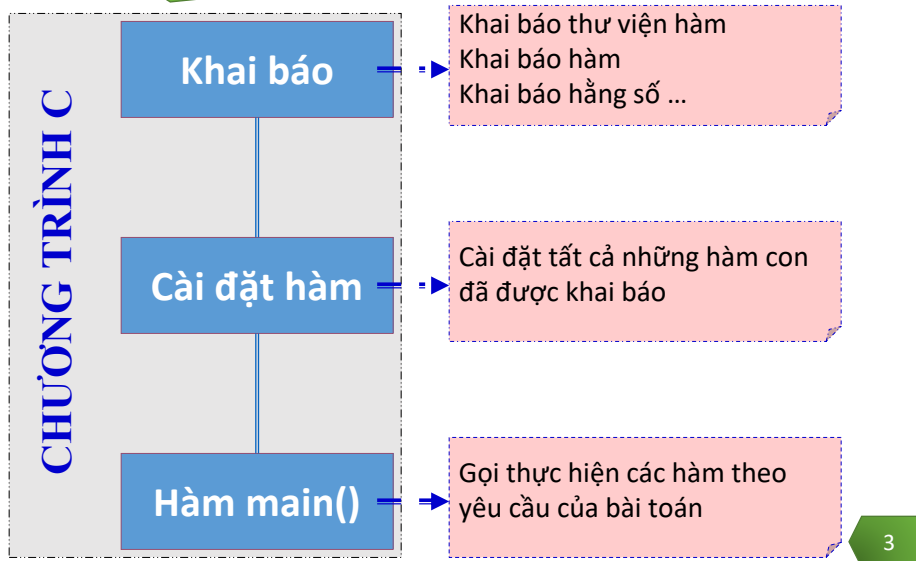
## Bài 6. Chương trình con (6 tiết)

Biên soạn TS. Trần Minh Thái  
Giảng viên: Lê Thị Minh Nguyễn  
Email: nguyentm@huflit.edu.vn

### Mục tiêu

1. Trình bày kỹ thuật lập trình bằng chương trình con
2. Cách thức phân tích bài toán thành các chương trình con

## Cấu trúc chương trình



## Khái niệm

Hàm (chương trình con - subroutine) là một khối lệnh, **thực hiện trọn vẹn một công việc nhất định** (module), được đặt tên và được gọi thực thi nhiều lần tại nhiều vị trí

Khi nào sử dụng hàm?

1. Khi có một công việc giống nhau cần thực hiện ở nhiều vị trí
2. Khi cần chia nhỏ chương trình để dễ quản lý

## Khái niệm

- Hàm có thể được gọi từ chương trình chính (hàm main) hoặc từ 1 hàm khác
- Hàm có giá trị trả về hoặc không
- Nếu hàm không có giá trị trả về gọi là thủ tục (*procedure*)

5

## Khái niệm

- *Hàm thư viện*: là những hàm đã được xây dựng sẵn. Muốn sử dụng các hàm thư viện phải khai báo thư viện chứa nó trong phần khai báo `#include`
- *Hàm do người dùng định nghĩa*

6

## Mẫu hàm

**<KDL trả về của hàm> TênHàm([ds tham số]);**

Kiểu dữ liệu trả về của hàm (kết quả của hàm/ đầu ra), gồm:

- **void**: Không trả về giá trị
- **float / int / long / char \*/ kiểu cấu trúc / ...** : Trả về kết quả tính được với KDL tương ứng

7

## Mẫu hàm

- **TênHàm**: Đặt tên theo qui ước sao cho phản ánh đúng chức năng thực hiện của hàm
- **Danh sách các tham số (nếu có)**: đầu vào của hàm

*Trong một số trường hợp có thể là đầu vào và đầu ra của hàm nếu kết quả đầu ra có nhiều giá trị - Tham số này gọi là tham chiếu, phần này sẽ đề cập sau*

8

## Hàm không trả về giá trị

### Cài đặt

```
void TênHàm([danh sách các tham số])
{
    Khai báo các biến cục bộ
    Các câu lệnh / khối lệnh hay lời gọi đến hàm khác.
}
```

### Gọi hàm

TênHàm(danh sách tên các đối số);

Những phương thức loại này thường rơi vào những **nhóm chức năng: Nhập / xuất dữ liệu, thống kê, sắp xếp, liệt kê**

9

## Ví dụ

Viết chương trình nhập số nguyên dương  $n$  và in ra màn hình các ước số của  $n$

Phân tích bài toán:

- **Input:**  $n$  (Để xác định tham số)  
KDL: số nguyên dương (*int*).
- **Output:** In ra các ước số của  $n$  (KDL trả về của hàm)  
Xuất ra màn hình → Không trả về giá trị → KDL của hàm là ***void***
- **Xác định tên hàm:** Hàm này dùng in ra các ước số của  $n$  nên có thể đặt là *LietKeUocSo*

*void LietKeUocSo(int n);*

10

```
#include <stdio.h>

void LietKeUocSo(int n)
{
    for (int i = 1; i <= n; i++)
    {
        if (n % i == 0)
            printf("%d\t", i);
    }
}

int main()
{
    int n;
    printf("Nhap so nguyen duong n: ");
    scanf("%d", &n);
    printf("Cac uoc so cua %d la: ", n);
    LietKeUocSo(n);

    return 0;
}
```

Muốn thực thi hàm thì  
phải gọi hàm

11

## Bài tập

Viết chương trình nhập vào **một số nguyên dương  $n$  gồm  $k$  chữ số ( $k \geq 1$ )**, in ra màn hình **các chữ số** sao cho mỗi chữ số xuất trên một dòng.

**Hướng dẫn: viết các hàm sau**

1. Hàm in ra màn hình các chữ số của số nguyên dương  $n$  gồm  $k$  chữ số (*Mỗi chữ số in trên 1 dòng*)
2. Hàm **main()** để nhập vào một số nguyên  $n$  gồm  $k$  chữ số, gọi hàm in ra màn hình các chữ số của  $n$

12

## Hàm có trả về giá trị

### Cài đặt

```
<KDL trả về> TênHàm([danh sách các tham số])
{
    <Kiểu dữ liệu trả về> kq;
    Khai báo các biến cục bộ
    Các câu lệnh / khối lệnh hay lời gọi đến hàm khác.
    return kq;
}
```

13

## Hàm có trả về giá trị

### Gọi hàm

<KDL trả về của hàm> Tên biến = TênHàm (danh sách tên các đối số);

Những hàm này thường rơi vào các nhóm: **Tính tổng, tích, trung bình, đếm, kiểm tra, tìm kiếm**

14

## Ví dụ

Viết chương trình nhập số nguyên dương  $n$  và tính tổng

$$S_n = 1 + 2 + 3 + \dots + n \quad ; n > 0$$

Phân tích bài toán:

**Input:**  $n$  (Tham số)

- KDL: số nguyên dương (*int*).

**Output:** Tổng  $S$  (KDL phương thức)

- Trả về giá trị của  $S$ .
- $S$  là tổng các số nguyên dương nên  $S$  cũng là số nguyên dương  $\rightarrow$  KDL trả về của hàm là *int* (hoặc *long*).

**TênHàm:** Dùng tính tổng  $S$  nên có thể đặt là *TinhTong*

*int TinhTong(int n);*

15

```
#include <stdio.h>

int TinhTong(int n)
{
    int kq = 0;
    for (int i = 1; i <= n; i++)
        kq += i;
    return kq;
}

int main()
{
    int n, s;
    printf("Nhap vao so nguyen n: ");
    scanf("%d", &n);
    s = TinhTong(n);
    printf("Tong tu 1 den n = %d\n", s);

    return 0;
}
```

Lưu giá trị trả về của  
hàm TinhTong

16



## Bài tập

Viết chương trình nhập vào 3 số nguyên: gio, phut, giay. Kiểm tra xem gio, phut, giay có tạo thành giờ phút giây hợp lệ không? In kết quả kiểm tra ra màn hình.

**Hướng dẫn: viết các hàm sau**

1. Hàm kiểm tra gio, phut, và giay cho trước có hợp lệ không?
2. Sử dụng hàm **main()** để nhập vào 3 giá trị tương ứng: gio, phut, và giay. Gọi hàm kiểm tra và in kết quả ra màn hình.

17

## Bài tập

Nhập n, viết các hàm sau

1. Kiểm tra n có phải là số nguyên tố không
2. Hàm **main()**

18

## Bài tập

Viết chương trình nhập vào 3 số nguyên tương ứng ngày, tháng và năm. Kiểm tra xem ngày tháng năm có hợp lệ không? In kết quả ra màn hình.

19

**Tham số đầu ra (tham biến)**

## Vấn đề tham số - Ví dụ 1

Xét chương trình hoán vị 2 số nguyên a, b cho trước

21

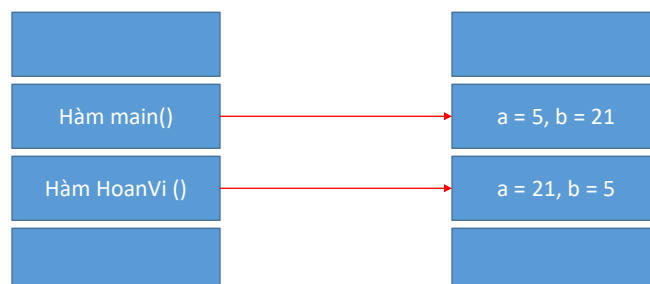
```
void HoanVi(int a, int b)
{
    int tam = a;
    a = b;
    b = tam;
    printf("Trong ham HoanVi: a = %d; b = %d\n", a, b);
}

int main()
{
    int a = 5, b = 21;
    printf("Truoc khi goi ham HoanVi: a = %d; b = %d\n", a, b);
    HoanVi(a, b);
    printf("Sau khi goi ham HoanVi: a = %d; b = %d\n", a, b);

    return 0;
}
```

22

## Kết quả



Trước khi gọi hàm HoanVi: a = 5; b = 21

Trong hàm HoanVi: a = 21; b = 5

Sau khi gọi hàm HoanVi: a = 5; b = 21

23

## Vấn đề tham số - Ví dụ 2

Tham trị (cục bộ)

```
void Nhap(int n)
{
    printf("Hay nhap vao so nguyen: ");
    scanf("%d", &n);
}
int main()
{
    int n = 0;
    Nhap(n);
    printf("Gia tri n sau khi gọi ham nhap: %d", n);
    getch();
    return 0;
}
```

Hay nhap vao so nguyen: 30  
Gia tri n sau khi gọi ham nhap: 0

24

## Vấn đề tham số - Ví dụ 2

Tham trị (cục bộ)

```
void Nhap(int &n)
{
    printf("Hay nhap vao so nguyen: ");
    scanf("%d", &n);
}
int main()
{
    int n = 0;
    Nhap(n);
    printf("Gia tri n sau khi goi ham nhap: %d", n);
    getch();
    return 0;
}
```

Hay nhap vao so nguyen: 30  
Gia tri n sau khi goi ham nhap: 30

25

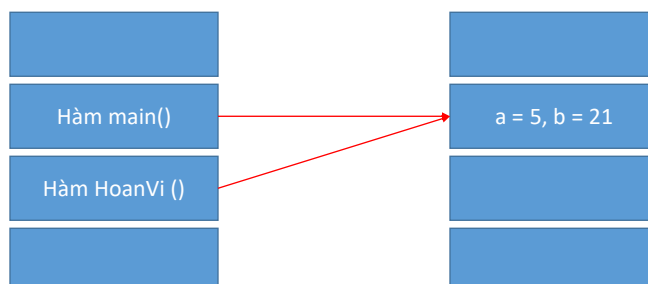
```
void HoanVi(int *a, int *b)
{
    int tam = a;
    a = b;
    b = tam;
    printf("Trong ham HoanVi: a = %d; b = %d\n", a, b);
}

int main()
{
    int a = 5, b = 21;
    printf("Truoc khi goi ham HoanVi: a = %d; b = %d\n", a, b);
    HoanVi(&a, &b);
    printf("Sau khi goi ham HoanVi: a = %d; b = %d\n", a, b);

    return 0;
}
```

26

## Kết quả



*Thay đổi giá trị  $a, b$  của hàm `HoanVi()` cũng là  $a, b$  của hàm `main()`*

Trước khi gọi hàm `HoanVi`:  $a = 5; b = 21$   
 Trong hàm `HoanVi`:  $a = 21; b = 5$   
 Sau khi gọi hàm `HoanVi`:  $a = 21; b = 5$

27

## Tầm vực của biến

- Phạm vi khối
- Phạm vi hàm
- Phạm vi tập tin
- Phạm vi chương trình

28

## Phạm vi khối

- Một khối được giới hạn bởi ngoặc {}
- Biến khai báo trong khối có phạm vi khối, nghĩa là nó chỉ hoạt động trong khối đó
- Phạm vi này còn gọi là **cục bộ**, và biến được gọi là **biến cục bộ** (*local variable*)

29

## Phạm vi khối (tt)

```
int main()
{
    int i = 20;
    {
        int i = 10;
        printf("Gia tri i ben trong khoi: %d\n", i);
    }
    printf("Gia tri i ben ngoai khoi: %d", i);
    getch();
    return 0;
}
```

### Kết quả

Gia tri i ben trong khoi: 10  
 Gia tri i ben ngoai khoi: 20

30

## Phạm vi hàm

Hoạt động từ đầu đến cuối một hàm, chỉ có tác dụng trong hàm

```
int main()
{
    int k;
    float m;
    double x;
    //Các lệnh khác
    //...
    getch();
    return 0;
}
```

31

## Phạm vi tập tin

Biến được khai báo toàn cục và có kèm từ khóa **static**

```
static int y = 0;
static float z = 0.0;
int main()
{
    int i;
    //Các lệnh
    getch();
    return 0;
}
```

32



## Phạm vi chương trình

- Được khai báo bên ngoài hàm  
→ **biến toàn cục** (*global variable*)
- Không nên “lạm dụng” biến toàn cục nếu không thực sự cần thiết

```
int a, b;
void Nhap()
{
    printf("Nhap a: ");
    scanf("%d", &a);
    printf("Nhap b: ");
    scanf("%d", &b);
}
int main()
{
    int c;
    Nhap();
    c = a + b;
    printf("Tong = %d", c);
    getch();
    return 0;
}
```

33

## Nguyên tắc xây dựng hàm

- **Kết quả của hàm?** → KDL trả về của hàm
- Hàm làm gì? → Xác định tên hàm
- **Hàm cần những thông tin gì truyền vào?** → Tham số  
Ứng với mỗi tham số đã xác định, xác định xem tham số đã có giá trị trước khi truyền vào hàm chưa?
  - Nếu chưa có → **Tham chiếu**
  - Nếu có, xét 2 trường hợp sau
    - + Sau khi thực hiện xong vẫn không đổi → **Tham trị**
    - + Sau khi thực hiện xong bị thay đổi → **Tham chiếu**

34

## Bài tập ví dụ

Viết chương trình tính diện tích và chu vi hình tròn với bán kính là số nguyên được nhập từ bàn phím

*Chương trình gồm các yêu cầu:*

1. *Nhập bán kính  $r$  của hình tròn sao cho  $r > 0$*
2. *Tính chu vi hình tròn*
3. *Tính diện tích hình tròn*
4. *Xuất kết quả*

35

## Bài tập ví dụ

### ***Hàm nhập bán kính hình tròn***

- Đầu vào: bán kính  $r$  (số nguyên), sau khi nhập xong  $r$  thì giá trị được nhập và hàm gọi thực hiện phải có cùng giá trị

→  *$r$  là tham chiếu*

- Đầu ra: không có giá trị trả về → *KDL trả về của hàm là **void***

→ Nguyên mẫu hàm: ***void NhapBanKinh(int \*r);***

36

## Bài tập ví dụ

### *Hàm tính chu vi hình tròn*

- Đầu vào: bán kính  $r$  của hình tròn
- Đầu ra: chu vi của hình tròn (số thực – do nhân với  $\pi$  là số thực) → **KDL trả về của hàm là float**
- Nguyên mẫu hàm: ***float TinhChuVi(int r);***

37

## Bài tập ví dụ

### *Hàm tính diện tích hình tròn*

- Đầu vào: bán kính  $r$  của hình tròn
- Đầu ra: diện tích của hình tròn (số thực – do nhân với  $\pi$ )
- **KDL trả về của hàm là float**
- Nguyên mẫu hàm: ***float TinhDienTich(int r);***

38

## Bài tập ví dụ

### *Hàm xuất kết quả*

- Đầu vào: bán kính  $r$  của hình tròn
- Đầu ra: xuất chu vi và diện tích của hình tròn → Không trả về giá trị → **KDL trả về của hàm là void**
- Nguyên mẫu hàm: **`void XuatKetQua(int r);`**

39

## Khai báo thư viện và nguyên mẫu hàm

```
#include<stdio.h>
#include<conio.h>

void NhapBanKinh(int *r);
float TinhChuVi(int r);
float TinhDienTich(int r);
void XuatKetQua(int r);
```

40

## Cài đặt hàm

```
void NhapBanKinh(int *r)
{
    do{
        printf("Nhap vao ban kinh r: ");
        scanf("%d", r);
        if(*r<=0)
            printf("Nhap lai r (r>0)!!!\n");
    }While(*r<=0);
}
float TinhChuVi(int r)
{
    return 2 * 3.14f *r;
}
float TinhDienTich(int r)
{
    return 3.14f*r*r;
}
```

41

## Cài đặt hàm

```
void XuatKetQua(int r)
{
    float cv = TinhChuVi(r);
    float dt = TinhDienTich(r);
    printf("Chu vi hinh tron ban kinh %d = %f\n", r, cv);
    printf("Dien tich hinh tron ban kinh %d = %f", r, dt);
}
```

42

## Cài đặt hàm main()

```
int main()
{
    int r;
    NhapBanKinh(&r);
    XuatKetQua(r);
    getch();
    return 0;
}
```

43

## Bài tập

- (1) Viết chương trình tính diện tích và chu vi của hình chữ nhật với chiều dài và chiều rộng là 2 số nguyên dương được nhập từ bàn phím.
- (2) Viết chương trình nhập 2 số nguyên a, b (khác 0). Tìm USCLN và BSCNN của hai số nguyên đó

44

## Bài tập

(3) Nhập vào 3 số nguyên  $a, b, c$ . Kiểm tra xem chúng có thành lập thành 3 cạnh của một tam giác hay không? Nếu có, hãy tính diện tích, chiều dài mỗi đường cao và in kết quả ra màn hình.

- Diện tích hình tam giác:  $s = \sqrt{p(p-a)(p-b)(p-c)}$

với  $p$  là nửa chu vi của tam giác

- Đường cao:  $h_a = 2s/a, h_b = 2s/b, h_c = 2s/c$

45

## Bài tập

(4) Viết chương trình nhập số nguyên dương  $n$ , tính tổng các ước số của  $n$ .

Ví dụ: Nhập  $n=6$

*Tổng các ước số từ 1 đến  $n$ :  $1+2+3+6=12$ .*

46

## Bài tập

(5) Viết chương trình tính tiền thuê máy dịch vụ Internet.

Với dữ liệu nhập vào là giờ bắt đầu thuê (GBD), giờ kết thúc thuê (GKT), mã số máy thuê (SoMay).

- Với:  $6 \leq \text{GBD} < \text{GKT} \leq 21$  (Giờ là số nguyên).
- Đơn giá: 2500đ cho mỗi giờ máy trước 17:30 và 3000đ cho mỗi giờ máy sau 17:30.

47

## Bài tập

(6) Viết chương trình tính tiền lương ngày cho công nhân, cho biết trước giờ vào ca, giờ ra ca của mỗi người.

**Giả sử rằng:**

- Tiền trả cho mỗi giờ trước 12 giờ: 6000đ và sau 12 giờ: 7500đ.
- Giờ vào ca sớm nhất là 6 giờ sáng và giờ ra ca trễ nhất là 18 giờ (Giờ là số nguyên).

48



Q&A



49