CHƯƠNG 5: CON TRỎ, CHƯƠNG TRÌNH CON VÀ ĐỐI TƯỢNG

KIỂU CON TRỎ

- Kiểu con trỏ là một lớp các đtdl mà giá trị dữ liệu của nó là vị trí của một đtdl khác.

Pascal: type VECT = array [1..10] of integer;

var p: ^VECT

C: int *p;

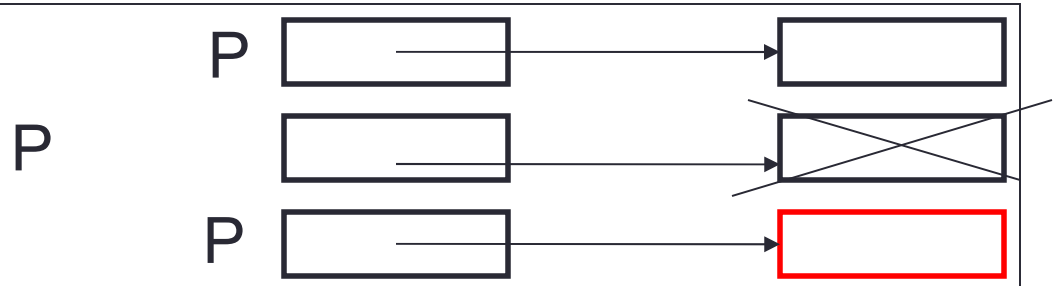
Con trỏ: - chỉ trỏ đến các đtdl cùng 1 kiểu

- có thể trỏ đến những đtdl thuộc những kiểu khác nhau

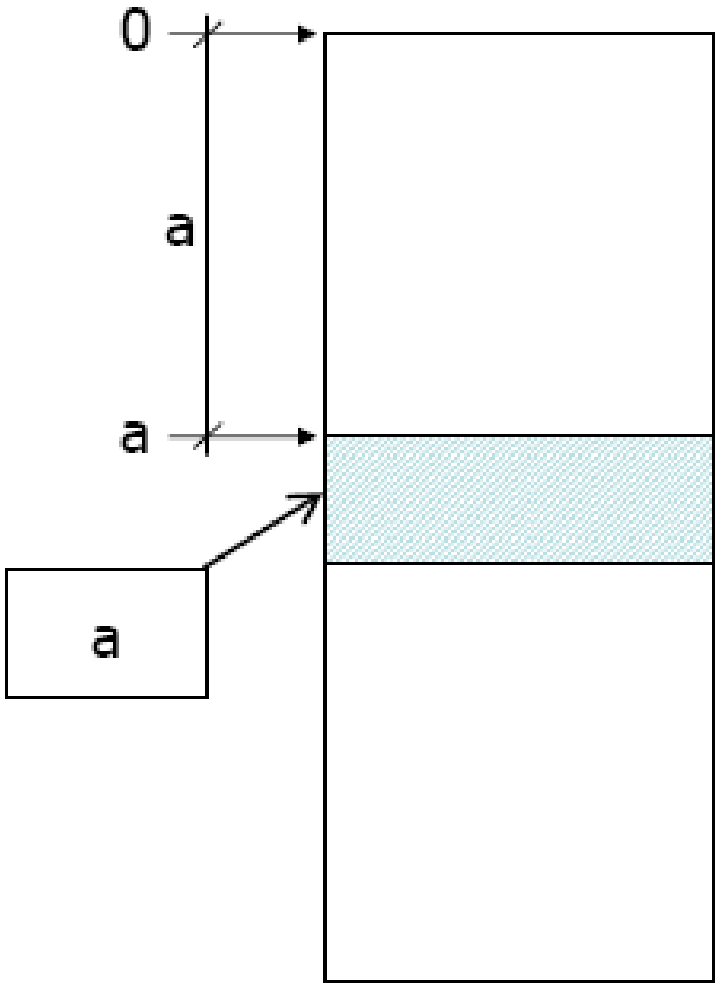
- C ?? Pascal ??

TÁC VỤ

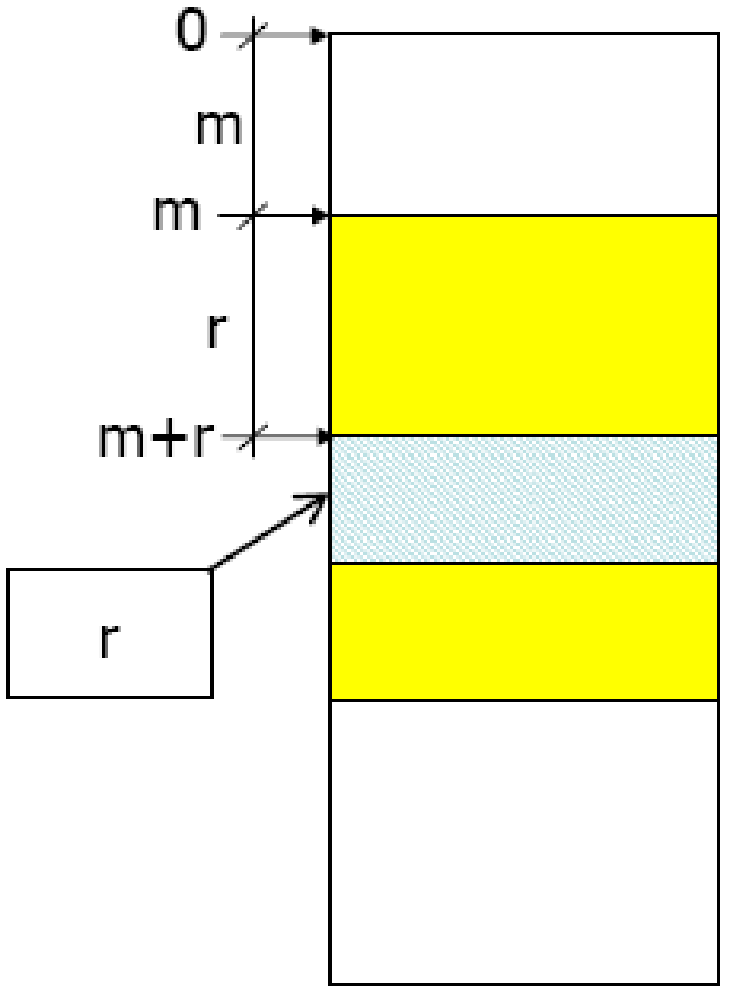
- Khởi tạo: new(p)
- Hủy bỏ: dispose(p)
- Tham khảo: ^p
- Lấy địa chỉ: p = &x



LƯU TRỮ



Địa chỉ tuyệt đối



Địa chỉ tương đối

CHƯƠNG TRÌNH CON

- Chương trình con là tác vụ trừu tượng được định nghĩa bởi người lập trình.
- Đặc tả chương trình con bao gồm
  - Tên chương trình con
  - Số lượng, thứ tự và kiểu của các đối số
  - Số lượng, thứ tự và kiểu của các kết quả
  - Hành vi của chương trình
- Hiện thực của chương trình con bao gồm các khai báo dữ liệu cục bộ và các phát biểu trong thân ct con.

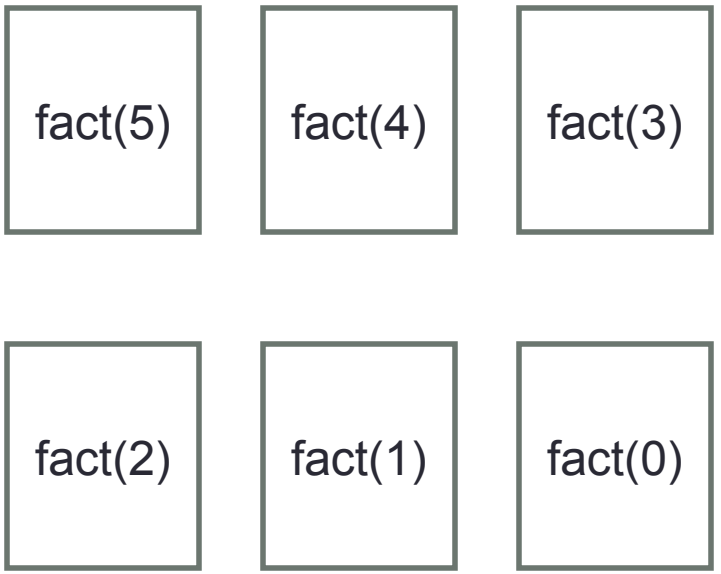
```
function FUNC(x: real; y: integer): real;      // đặc tả
var a: array [1..10] of real;                  //khai báo dữ liệu cục bộ
begin ... end                                  // các phát biểu
```

ĐỊNH NGHĨA VÀ BẢN HOẠT ĐỘNG

- Định nghĩa chương trình con

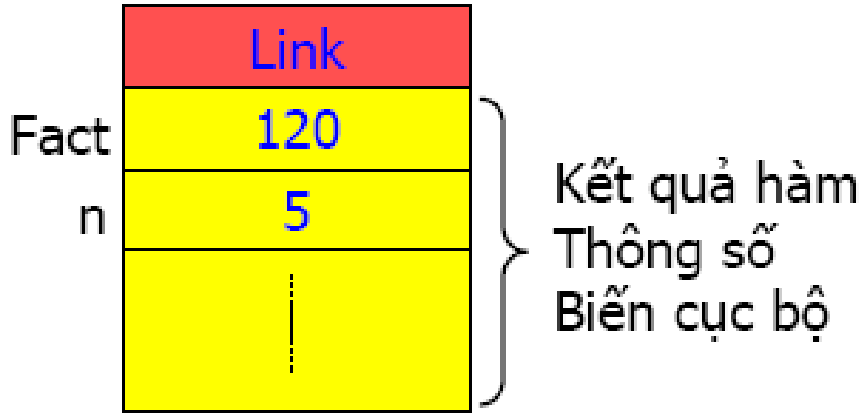
```
function fact(n: integer): integer;
begin
    if (n = 0) fact := 1
    else fact := n * fact (n -1);
end
```

Bản hoạt động chương trình con

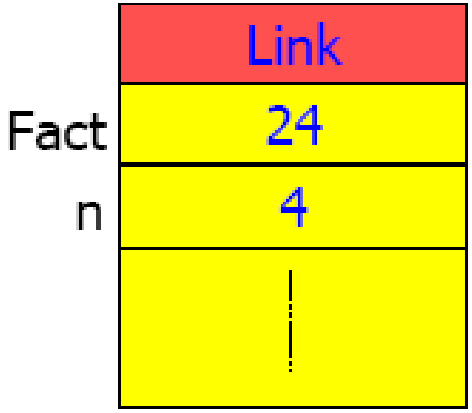


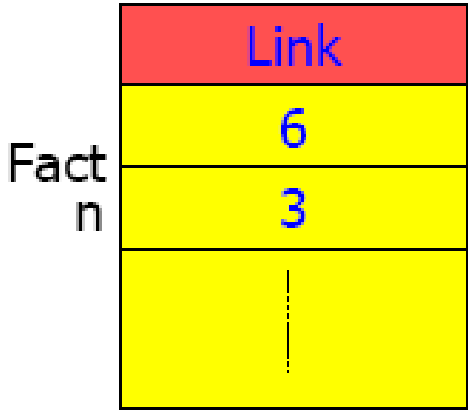
HIỆN THỰC BẢN HOẠT ĐỘNG

Phân đoạn mã



Bản ghi hoạt động





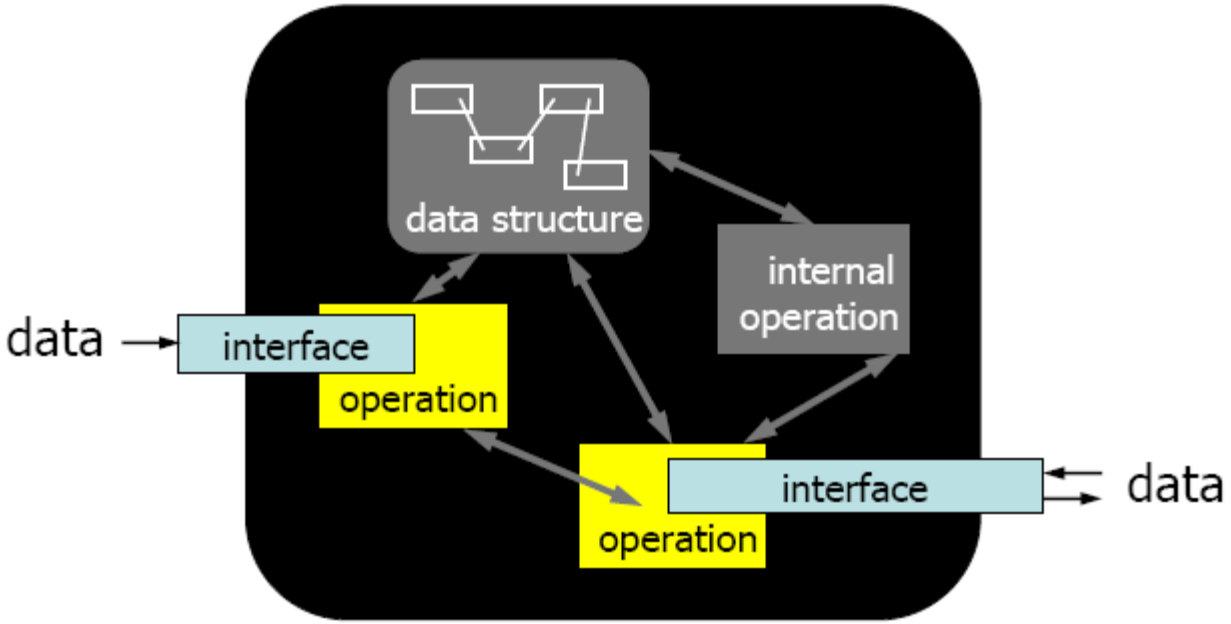
KIỂU ĐỐI TƯỢNG

- Một đối tượng bao gồm cả dữ liệu (thuộc tính) và tác vụ (phương thức).
- Đóng gói dữ liệu và tác vụ (encapsulation)

```
class A {
    attribute a1;
    attribute a2;
    method m1;
    method m2;
}
```

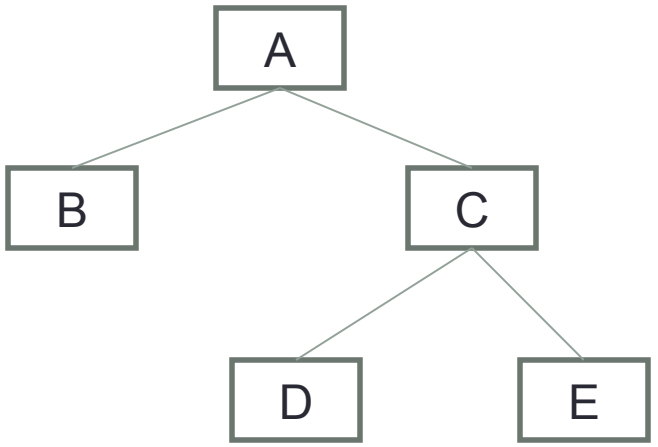
KIỂU ĐỐI TƯỢNG

Che giấu thông tin



KIỂU ĐỐI TƯỢNG

- Sự phân cấp và tính thừa kế
  - Các lớp được tổ chức phân cấp
  - Giữa các lớp có thể có sự thừa kế các thuộc tính và phương thức




```
class A D E                    class B extends A
{ attribute a1;                { attribute a2;
  method m1;                     method m2;
  method m2;                     method m3;
}                              }
```

- Tính đa hình (polymorphism)

```
B x; ...
x.m2();
```

HIỆN THỰC KIỂU ĐỐI TƯỢNG

```
class A {
  attribute a1;
  attribute a2;
  method m1;
  method m2;
}
```