

Chương 06. Hàm tạo và hàm hủy (constructor & destructor)

I. Giới thiệu về hàm tạo và hàm hủy

I.1. Giới thiệu về hàm tạo và hàm hủy

I.2. Hàm tạo và hàm hủy do người lập trình viết

II. Hàm tạo có đối số

II.1. Hàm tạo hai đối số

II.2. Hàm tạo mặc định

II.3. Hàm tạo một đối số

III. Hàm tạo sao chép

Điểm đặc biệt của hàm tạo, hàm hủy

- Không cần khai báo, định nghĩa (viết) vẫn có
- Tự động thực hiện, không cần lời gọi hàm
- Hàm tạo có tên hàm trùng với tên lớp, hàm hủy có tên hàm là tên lớp với ký tự ~ đứng trước. Ví dụ: SoPhuc(), ~SoPhuc()
- Không có kiểu trả về, kể cả kiểu void

Khi nào phải viết hàm tạo, hàm hủy

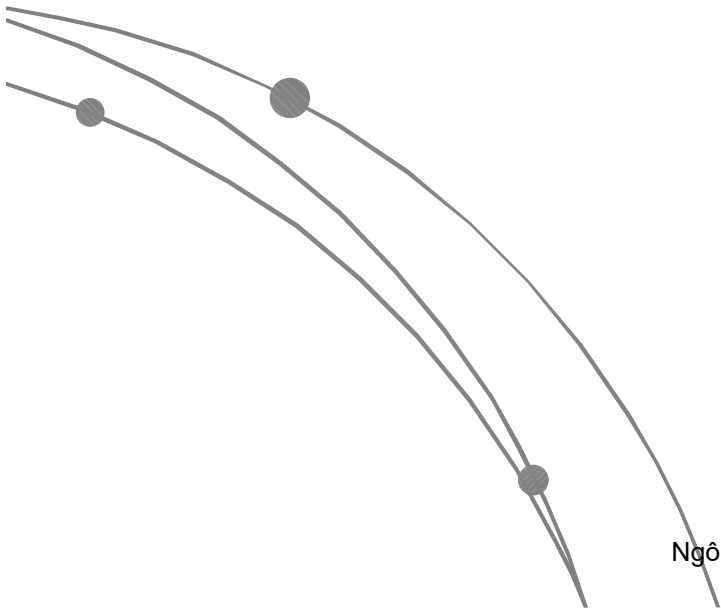
- Bình thường thì ta không phải viết hàm tạo, hàm hủy vì trình biên dịch sẽ tự động thêm vào lớp.
- Ta chỉ phải viết hàm tạo, hàm hủy trong 2 tình huống sau:
 - 1) Khi cần khởi tạo giá trị ban đầu cho biến của đối tượng thì phải viết hàm tạo.
 - 2) Khi trong đối tượng có sử dụng con trỏ và biến động thì phải viết cả hàm tạo và hàm hủy.

Chức năng

- Hàm tạo có chức năng tạo đối tượng và cấp phát tài nguyên cho đối tượng. Hàm hủy có chức năng hủy đối tượng và giải phóng tài nguyên chiếm giữ bởi đối tượng.
- Trong một lớp thường có nhiều hàm tạo với đối số khác nhau và có một hàm hủy không đối số. Mỗi hàm tạo cho ta một cách tạo đối tượng. Mỗi đối tượng chỉ được tạo bởi một hàm tạo.

Hàm tạo mặc định

- Là hàm tạo không đối số
- Trình biên dịch chỉ thêm vào lớp hàm tạo không đối số.



Hàm tạo có đối số

- Làm thế nào để truyền được đối số cho hàm tạo khi không có lời gọi hàm?
- Truyền đối số bằng cách để danh sách đối số trong ngoặc đơn ngay sau tên đối tượng khi tạo đối tượng.

Ví dụ: Lớp số phức có hàm tạo 2 đối số. Tạo đối tượng số phức p bằng hàm tạo 2 đối số như sau: `SoPhuc p(2, 5);`

Hàm tạo một đối số

- Hàm tạo một đối số đặc biệt hơn hàm tạo có đối số khác ở hai điểm sau:
 - Khi truyền đối số có thể dùng ký pháp khởi tạo là dấu `=`. Ví dụ: `SoPhuc p=2;`
 - Hàm tạo 1 đối số tự động thực hiện khi gán 1 giá trị có kiểu đối số cho đối tượng.
Ví dụ: `p = 5;`
 \Rightarrow Hàm tạo 1 đối số còn được gọi là hàm chuyển đổi kiểu dữ liệu.

Hàm tạo sao chép

- Là hàm tạo một đối số nhưng đối số lại chính là đối tượng cùng lớp với hàm tạo. Hàm tạo sao chép có chức năng tạo ra đối tượng mới là bản sao của đối tượng đã có.
- Hàm tạo sao chép đặc biệt hơn các hàm một đối số ở điểm sau:
 - Tự động thực hiện khi truyền đối tượng cho hàm theo giá trị.
 - Tự động thực hiện khi hàm trả về đối tượng

Hàm tạo sao chép

- Vì 2 điểm đặc biệt này mà hàm tạo sao chép luôn được trình biên dịch tự động thêm vào lớp cùng với hàm tạo không đối số.
- Bình thường ta không phải viết hàm tạo sao chép, ta chỉ phải viết khi trong đối tượng có sử dụng con trỏ và biến động.

I.1. Giới thiệu về hàm tạo và hàm hủy

- Hàm tạo và hàm hủy là hai hàm thành viên đặc biệt của đối tượng. Hàm tạo được thực hiện tự động khi đối tượng được tạo, còn hàm hủy được thực hiện tự động khi đối tượng bị hủy.
- Chúng ta thường viết hàm tạo để khởi tạo đối tượng, viết hàm hủy để giải phóng bộ nhớ cấp phát bởi hàm tạo.
- Dù người lập trình có viết hay không viết hàm tạo và hàm hủy thì trình biên dịch vẫn tạo ra những mã lệnh để tạo đối tượng, cấp phát bộ nhớ cho nó và thực hiện một số khởi tạo nào đó; tạo ra mã lệnh để giải phóng bộ nhớ chiếm bởi đối tượng và thực hiện nhiều hoạt động dọn dẹp khác khi đối tượng bị hủy.

Chương 4. Hàm tạo và hàm hủy

I. Giới thiệu về hàm tạo và hàm hủy

I.1. Giới thiệu về hàm tạo và hàm hủy

I.2. Hàm tạo và hàm hủy do người lập trình viết

II. Hàm tạo có đối số

II.1. Hàm tạo hai đối số

II.2. Hàm tạo mặc định

II.3. Hàm tạo một đối số

III. Hàm tạo sao chép

I.2. Hàm tạo và hàm hủy do người lập trình viết

- Người lập trình có thể tự định nghĩa hàm tạo và hàm hủy của riêng mình.
- Hàm tạo và hàm hủy có thể định nghĩa ngay trong mô tả lớp. Cả hai hàm này đều không có kiểu trả về, kể cả kiểu void. Hàm tạo có tên trùng với tên lớp, hàm hủy cũng có tên trùng với tên lớp nhưng có dấu ~ đứng trước.
- Ví dụ: Định nghĩa một lớp chỉ có hàm tạo và hàm hủy, sau đó tạo 2 đối tượng của lớp này và xem các hàm tạo và hàm hủy thực hiện thế nào.

I.2. Hàm tạo và hàm hủy do người lập trình viết (tiếp)

- Ta có thể dùng hàm tạo để khởi tạo giá trị cho các biến của đối tượng. Có 2 cách khởi tạo:
 - Dùng danh sách khởi tạo: danh sách khởi tạo nằm cùng dòng với tên hàm tạo, bắt đầu bằng dấu hai chấm, sau đó là các biến cần khởi tạo cách nhau bởi dấu chấm, giá trị khởi tạo đặt trong ngoặc đơn ngay sau tên biến. Ví dụ: giả sử lớp Alpha có 2 biến nguyên là a và b, hàm tạo khởi tạo giá trị cho 2 biến này như sau:

Alpha() : a(5), b(6)

{ }

- Dùng lệnh gán giá trị trong thân của hàm tạo. Cách này chỉ áp dụng với một số biến lớn như biến mảng, đối tượng.
- Ví dụ:

Chương 4. Hàm tạo và hàm hủy

I. Giới thiệu về hàm tạo và hàm hủy

I.1. Giới thiệu về hàm tạo và hàm hủy

I.2. Hàm tạo và hàm hủy do người lập trình viết

II. Hàm tạo có đối số

II.1. Hàm tạo hai đối số

II.2. Hàm tạo mặc định

II.3. Hàm tạo một đối số

III. Hàm tạo sao chép

II.1. Hàm tạo hai đối số

- Ví dụ: Viết lớp số phức có 1 hàm tạo không đối số để khởi tạo phần thực và phần ảo bằng 0, có 1 hàm tạo hai đối số để khởi tạo phần thực và phần ảo bằng đối số, có một hàm hủy.
- Bài tập về nhà: Viết một lớp Stack có thể chứa các số nguyên. Nhập vào 1 số nguyên dương, đưa ra số nhị phân.

II.1. Hàm tạo hai đối số (tiếp)

- Cú pháp “Gọi” hàm tạo có hai đối số: Bởi vì hàm tạo được thực hiện tự động nên ta không thể sử dụng lời gọi hàm thông thường để truyền đối số cho nó. Việc truyền đối số cho hàm tạo được thực hiện khi tạo đối tượng. Giá trị của các đối số được đặt trong ngoặc đơn sau tên đối tượng.

Tên_lớp Tên_đối_tượng(Danh sách đối số);

Chương 4. Hàm tạo và hàm hủy

I. Giới thiệu về hàm tạo và hàm hủy

I.1. Giới thiệu về hàm tạo và hàm hủy

I.2. Hàm tạo và hàm hủy do người lập trình viết

II. Hàm tạo có đối số

II.1. Hàm tạo hai đối số

II.2. Hàm tạo mặc định

II.3. Hàm tạo một đối số

III. Hàm tạo sao chép

II.2. Hàm tạo mặc định

- Nếu ta không định nghĩa hàm tạo thì trình biên dịch sẽ tạo ra một hàm tạo mặc định, hàm tạo này không có đối số.
- Tuy nhiên, nếu ta tự định nghĩa hàm tạo, dù là có đối số hay không có đối số thì trình biên dịch sẽ không tạo ra hàm tạo mặc định nữa.
- Bài tập 2: Nếu trong lớp `airtime` ta định nghĩa hàm tạo 2 đối số thì khai báo sau sẽ sinh ra lỗi.
`airtime t1, t2;`
Hãy thay đổi lớp `airtime` để khai báo trên vẫn đúng trong khi lớp vẫn có hàm tạo 2 đối số.

Chương 4. Hàm tạo và hàm hủy

I. Giới thiệu về hàm tạo và hàm hủy

I.1. Giới thiệu về hàm tạo và hàm hủy

I.2. Hàm tạo và hàm hủy do người lập trình viết

II. Hàm tạo có đối số

II.1. Hàm tạo hai đối số

II.2. Hàm tạo mặc định

II.3. Hàm tạo một đối số

III. Hàm tạo sao chép

II.3. Hàm tạo một đối số

- Hàm tạo một đối số có vai trò quan trọng trong lập trình hướng đối tượng. Nó được dùng để chuyển đổi một đối tượng lớp này sang lớp khác. Việc chuyển đổi này thường dùng với các lớp biểu diễn kiểu dữ liệu. Chính vì lý do này mà hàm tạo con được gọi là hàm chuyển đổi.
- Để thấy được cách khai báo và sử dụng hàm tạo một đối số ta xét ví dụ sau: Giả sử lớp TypeA có hàm tạo một đối số kiểu int được khai báo như sau: (Trang sau)

II.3. Hàm tạo một đối số (tiếp)

```
class TypeA
{
    public:
        TypeA(int i) //Ham tao mot doi so
        {
            //Thuc hien chuyen doi gia tri int toi gia tri TypeA
        }
};

void main()
{
    int b=50; //Khai bao mot bien int co gia tri 50

    TypeA ta1(b); //Khoi tao doi tuong ta1 bang gia tri int
    //TypeA ta1=b; //Tuong duong voi ta1(b)
}
```

II.3. Hàm tạo một đối số (tiếp)

- Dấu = trong cách thứ hai không phải là toán tử gán, nó chỉ có tác dụng gọi hàm tạo một đối số.
- Bài tập về nhà: Hãy xây dựng một lớp về chuỗi ký tự trong đó có sử dụng hàm tạo một đối số để chuyển đổi một chuỗi ký tự thông thường thành đối tượng chuỗi ký tự.

Chương 4. Hàm tạo và hàm hủy

I. Giới thiệu về hàm tạo và hàm hủy

I.1. Giới thiệu về hàm tạo và hàm hủy

I.2. Hàm tạo và hàm hủy do người lập trình viết

II. Hàm tạo có đối số

II.1. Hàm tạo hai đối số

II.2. Hàm tạo mặc định

II.3. Hàm tạo một đối số

III. Hàm tạo sao chép

III. Hàm tạo sao chép

- Hàm tạo sao chép (copy constructor) cho phép ta tạo ra một đối tượng là bản sao của một đối tượng đã có.
- Hàm tạo sao chép là một hàm tạo chỉ có một đối số, đối số này là đối tượng của lớp chứa hàm tạo.

III. Hàm tạo sao chép (tiếp)

- Khi khai báo một biến kiểu cơ bản ta có thể khởi tạo giá trị cho nó.
Ví dụ:

```
int a=45;    //tao bien a va khoi tao bang 45
int b=a;    //tao bien b la ban sao cua a
//int b(a); //tuong duong voi lenh int b=a;
```

Dấu = trong khai báo trên không phải là lệnh gán, nó chỉ có nghĩa là khởi tạo.

- Với đối tượng ta cũng có thể tạo và khởi tạo cho nó giống như với một biến kiểu cơ bản. Ví dụ: giả sử có lớp Alpha

```
Alpha a1;    //tao mot doi tuong Alpha
a1.getdata();//dua du lieu vao doi tuong a1
Alpha a2=a1;//tao doi tuong a2 la ban sao cua a1, o day ham
//tao sao chep duoc goi de sao chep du lieu tu a1 sang a2
```

III. Hàm tạo sao chép (tiếp)

- Nếu không định nghĩa hàm tạo sao chép thì trình biên dịch sẽ tạo ra một hàm tạo sao chép mặc định. Hàm tạo sao chép mặc định sao chép y nguyên tất cả dữ liệu từ một đối tượng này sang một đối tượng khác. Bởi vậy, nếu ta chỉ cần sao chép dữ liệu từ đối tượng này sang đối tượng khác thì không cần định nghĩa hàm tạo sao chép. Còn nếu ta muốn làm những việc khác thì ta phải định nghĩa riêng một hàm tạo sao chép.
- Hàm tạo sao chép phải được khai báo để truyền đối số theo tham chiếu và đối số nên để là const.

III. Hàm tạo sao chép (tiếp)

- Hàm tạo sao chép được gọi trong các trường hợp sau:
 - Khi khởi tạo đối tượng
 - Khi truyền đối tượng cho hàm theo giá trị
 - Khi hàm trả về đối tượng
- Ví dụ về sử dụng hàm tạo, hàm hủy, biến ngoài và biến tự động: Viết chương trình tạo ra một lớp sao cho khi tạo đối tượng có thể khởi tạo đối tượng bằng một xâu ký tự; hàm tạo đối tượng cần hiển thị dữ liệu của đối tượng, đây là đối tượng thứ mấy và tổng số đối tượng đang tồn tại là bao nhiêu; còn hàm hủy cần hiển thị là hủy đối tượng thứ mấy và sau khi hủy thì còn bao nhiêu đối tượng đang tồn tại; trước khi hàm main kết thúc cần hiển thị tổng số đối tượng đã tạo ra trong thời gian chương trình chạy.

Bài tập

- Ứng dụng ngăn xếp lưu trữ kế tiếp để chuyển số nguyên dương hệ 10 sang hệ 2.
- Đọc dãy số nguyên từ tệp văn bản “songuyen.txt” đưa vào danh sách liên kết đơn. Các số nguyên trong DSLKD có thứ tự như trên tệp. Duyệt DSLKD để đưa dãy số nguyên ra màn hình.

IV-Mẫu lớp (Class Template)

- Mẫu lớp hay còn gọi là lớp tổng quát (Generic Class) là lớp sử dụng kiểu dữ liệu giả định, khi tạo đối tượng mới xác định kiểu dữ liệu cụ thể.
- Mẫu lớp cho phép tạo các đối tượng chứa kiểu dữ liệu khác nhau.
- Kiểu dữ liệu giả định trong mẫu lớp giống như tham số trong khai báo và định nghĩa hàm.

IV-Mẫu lớp (Class Template)

- Cú pháp khai báo mẫu lớp:

```
template <class T>
```

```
class TenLop
```

```
{
```

```
private:
```

```
public:
```

```
};
```

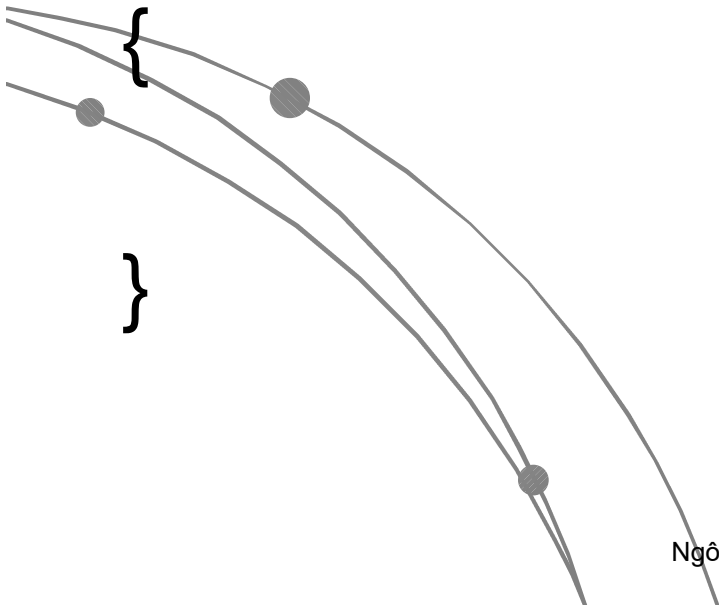
Trong đó T là tên
kiểu giả định tự đặt

IV-Mẫu lớp (Class Template)

- Cú pháp định nghĩa hàm thành viên ngoài khai báo mẫu lớp:

```
template <class T>
```

```
Kiểu TenLop<T>::TenHam(DS Tham số)
```

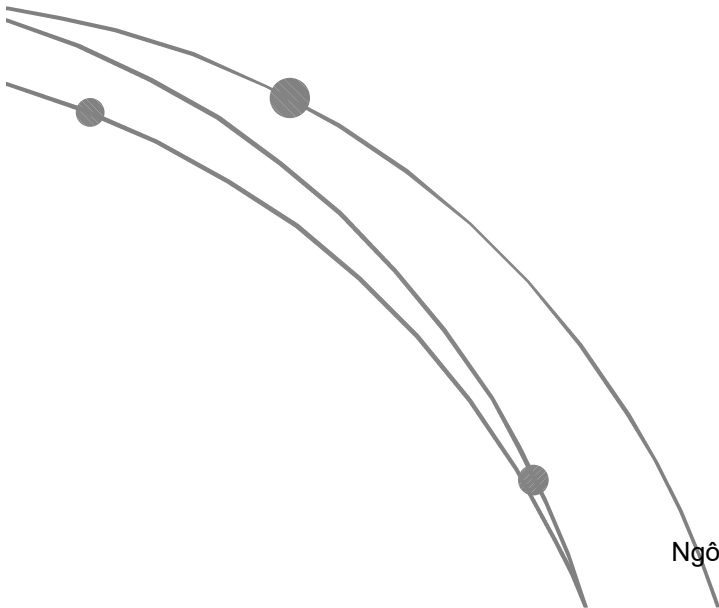


IV-Mẫu lớp (Class Template)

- Cú pháp tạo đối tượng từ mẫu lớp:

TenLop<Kiểu DL> TenDT;

Ví dụ: SList<int> List;



Bài tập

- Bài 1: Ứng dụng ngăn xếp lưu trữ kế tiếp chuyển số nguyên dương hệ 10 sang hệ 2 và 16. Yêu cầu sử dụng mẫu lớp trong cài đặt.
- Bài 2: Ứng dụng ngăn xếp lưu trữ phân tán chuyển số nguyên dương hệ 10 sang hệ 2.