BÀI TẬP THỰC HÀNH TUẦN 2

Nội dung:

- Hàm dựng, hàm dựng sao chép
- Hàm hủy
- Getter, Setter
- Bài tập tổng hợp cơ bản về lớp

Qui định nộp bài tập:

- Mỗi bài tập tương ứng với 1 project, tên Project là BaiXX, với XX là thứ tự của bài tập (Ví dụ bài tập 1 tên Project tương ứng là Bai02).

- Tất cả các bài tập được đặt bên trong một thư mục, tên thư mục theo qui định như sau: **BTH2_HoVaTen_MSSV**. Ví dụ Sinh viên Nguyễn Sơn Trà có MSSV là 19521269 thì đặt tên thư mục như sau: **BTH2_NguyenVanA_20521234**

- Sau đó nén thư mục trên thành tập tin **.zip hoặc .rar** (tên file nén cũng theo qui định như tên thư mục). Ví dụ **BTH2_NguyenVanA_20521234.rar**

- Lưu ý xóa thư mục được phát sinh sau khi biên dịch (thư mục Debug, .vs) của mỗi project

- Hình thức nộp bài: Nộp trên website môn học theo thời gian qui định.

- Những bài nộp không đúng qui định như trên sẽ không được chấm điểm (0 điểm)

- Tất cả các bài làm có tính chất sao chép (copy) sẽ nhận 0 điểm

1. Phương thức tạo lập - Constructor

- Phương thức tạo lập của một lớp đối tượng có nhiệm vụ thiết lập thông tin ban đầu cho các đối tượng thuộc về lớp ngay khi đối tượng được khai báo
- Nhiệm vụ của phương thức tạo lập để giúp cho lập trình viên tránh việc quên khởi tạo đối tượng trước khi sử dụng.
- Phương thức tạo lập là một phương thức đặc biệt được gọi thực hiện một cách tự động khi đối tượng thuộc lớp đó được khởi tạo
- Đặc điểm của phương thức tạo lập:
    - Phải có tên phương thức trùng với tên lớp
    - Không có kiểu trả về
    - Tự động chạy ngay khi đối tượng được tạo
    - Có khả năng chồng hàm (có nhiều phương thức tạo lập chồng nhau, overloading), phân biệt thông qua tham số truyền vào
- Phương thức tạo lập mặc định:
    - Không có tham số đầu vào
    - Nếu không được định nghĩa thì sẽ được trình biên dịch tự tạo ra trong trường hợp **chưa có bất kỳ phương thức tạo lập nào được định nghĩa**

```
//Default constructor với tử số là 0 và mẫu là 1
PhanSo()
{
	tuso = 0;
	mauso = 1;
}
```

    - Phương thức tạo lập mặc định được gọi khi khởi tạo đối tượng

```
PhanSo a; //default constructor gọi tại đây
cout << "Tu so la: " << a.LayTuSo() << endl;
cout << "Mau so la: " << a.LayMauSo() << endl;
```

```
Tu so la: 0
Mau so la: 1
```

- Phương thức tạo lập nhận tham số đầu vào
    - Phương thức tạo lập được định nghĩa với các tham số đầu vào khác nhau để khởi tạo dữ liệu cho đối tượng
    - Có thể có nhiều phương thức tạo lập theo cơ chế nạp chồng hàm (overloading), mỗi phương thức phân biệt qua tham số đầu vào khác nhau
    - Ví dụ cho phương thức tạo lập có 1 tham số đầu vào

```
//Constructor có 1 tham số đầu vào đại diện cho tử số
PhanSo(int tu)
{
	tuso = tu;
	mauso = 1;
}
```

- Ví dụ cho constructor có 2 tham số đầu vào:

```
//Constructor có 2 tham số, trong đó xử lý mẫu <= 0
PhanSo(int tu, int mau)
{
	//Mẫu < 0, đổi dấu tử và mẫu
	if (mau < 0)
	{
		tu = -tu;
		mau = -mau;
	}
	tuso = tu;
	//Mẫu số = 0, gán mặc định = 1
	if (mau == 0) mauso = 1;
	else mauso = mau;
}
```

- Phương thức tạo lập sao chép
  - Khởi tạo đối tượng dựa trên việc sao chép thông tin của một đối tượng sẵn có
  - Được **trình biên dịch cung cấp mặc định**, gọi là phương thức tạo lập sao chép mặc định (default copy constructor)
  - Có thể được định nghĩa và cài đặt lại

```
//Copy constructor
PhanSo(const PhanSo &temp)
{
	tuso = temp.tuso;
	mauso = temp.mauso;
}
```

  - Lưu ý, đầu vào của phương thức tạo lập mặc định là một lớp có chung kiểu, truyền tham chiếu (&) kết hợp với const với mục đích:
    - Không cho phép thay đổi đối tượng, chỉ mục đích sao chép (copy)
    - Nếu truyền tham trị, trình biên dịch sẽ tạo một biến tạm để copy giá trị vào giá trị tham số, dẫn đến việc gọi hàm copy, tạo thành một vòng lặp vô hạn.
  - Phương thức tạo lập sao chép được gọi trong trường hợp sau:

```
PhanSo a; //Default constructor
PhanSo b(a); //Copy constructor
PhanSo c = a;//Copy constructor
```

- Chuỗi phương thức tạo lập (constructor chaining)
  - Cho phép gọi lại phương thức tạo lập khác để chạy (constructor delegation)
  - Tránh việc xử lý lặp đi lặp lại nhiều lần trong lúc cài đặt constructor

```
//Constructor chaining - constructor delegation
//Gọi là constructor có 2 tham số, truyền vào 0 và 1
PhanSo(): PhanSo(0, 1) {}
//Constructor có 2 tham số, trong đó xử lý mẫu <= 0
PhanSo(int tu, int mau)
{
	//Mẫu < 0, đổi dấu tử và mẫu
	if (mau < 0)
	{
		tu = -tu;
		mau = -mau;
	}
	tuso = tu;
	//Mẫu số = 0, gán mặc định = 1
	if (mau == 0) mauso = 1;
	else mauso = mau;
}
```

2. Phương thức hủy – Destructor

- Phương thức hủy giúp dọn dẹp tài nguyên, thông tin bên trong đối tượng trước khi hủy giúp tránh xảy ra tình huống không mong muốn
- Phương thức hủy sẽ được gọi mặc định ngay khi đối tưởng không còn được sử dụng nữa và bị hủy, có nhiệm vụ thu hồi tài nguyên đã cấp
- Đặc điểm phương thức hủy:
  - Đặt tên trùng với tên lớp và có dấu ngã ~ đặt ở ngay phía trước
  - Không có tham số đầu vào
  - Không có giá trị trả về
  - Mỗi lớp chỉ có duy nhất một phương thức hủy
  - Chạy tự động khi đối tượng bị hủy
  - Chỉ gọi chạy một lần duy nhất
  - Phương thức hủy không cần cài đặt khi không có sử dụng vùng nhớ động

```
~PhanSo()  {}
```

3. Phương thức getter, setter

- Phương pháp lập trình hướng đối tượng có **tính đóng gói (Encapsulation)** để bao bọc và che giấu quá trình xử lý bên trong, ngoài ra còn ngăn cản không cho phép sự tác động trực tiếp vào thuộc tính. Do đó, lớp đối tượng cung cấp các phương thức getter và setter, những đối tượng khác chỉ có thể tác động thông qua những phương thức getter và setter này.
- Phương thức getter cho phép lấy thông tin thuộc tính của lớp đối tượng.
- Phương thức setter cho phép gán thông tin thuộc tính của lớp đối tượng.
- Phương thức getter và setter đều được đặt là public, trong đó các thuộc tính tương ứng sẽ được đặt là private

- Getter và setter đơn giản nhất là cho phép lấy và gán giá trị trực tiếp. Bên cạnh đó, còn có thể xử lý thông tin trước khi gán và lấy giá trị.
- Tham khảo ví dụ cho getter và setter của lớp đối tượng Phân số:
- Vậy bên ngoài muốn sử dụng thông tin hay thay đổi, bắt buộc phải gọi qua các phương thức getter và setter này, không thay đổi trực tiếp thuộc tính được do đã được gán tầm vực private
- Ví dụ về xử lý setter trong trường hợp đối tượng Phân số được gán mẫu số là 0:

```
//Setter cho mẫu số
void GanMauSo(int mauso)
{
	//Ví dụ về setter xử lý khi mẫu số bằng 0
	if (mauso == 0) cout << "Khong the gan gia tri mau so = 0\n";
	else this->mauso = mauso;
}
```

- Ví dụ về việc sử dụng getter và setter từ bên ngoài lớp đối tượng:

```
class PhanSo
{
//Thuộc tính có tầm vực là private, không cho thay đổi trực tiếp
private:
	int tuso;
	int mauso;

//Phương thức Getter và Setter là public, cho phép giao tiếp với
bên ngoài
public:
	//Getter cho tử số
	int LayTuSo()
	{
		//Bên trong có thể còn xử lý khác
		return tuso;
	}
	//Getter cho mẫu số
	int LayMauSo()
	{
		return mauso;
	}
	//Setter cho tử số
	void GanTuSo(int tuso)
	{
		this->tuso = tuso;
	}
	//Setter cho mẫu số
	void GanMauSo(int mauso)
	{
		this->mauso = mauso;
	}
};
```

```
void main()
{
	PhanSo a;
	int data;
	//Nhập tử số từ bàn phím
	cout << "Nhap tu so: ";
	cin >> data;
	//Gán tử số nhập vào thông qua setter
	a.GanTuSo(data);
	cout << "Nhap mau so: ";
	cin >> data;
	//Gán tử số nhập vào thông qua setter
	a.GanMauSo(data);
	//Lấy thông tin tử số mẫu số qua getter
	cout << "Tu so la: " << a.LayTuSo() << endl;
	cout << "Mau so la: " << a.LayMauSo() << endl;
}
```

```
Nhap tu so: 3
Nhap mau so: 4
Tu so la: 3
Mau so la: 4
```

4. Bài tập thực hành

1. Viết chương trình C++ cài đặt đối tượng Nhân viên, trong đó nhân viên được miêu tả có các thông tin sau:
   - Họ tên của nhân viên: kiểu chuỗi
   - Số ngày làm việc của nhân viên: kiểu số nguyên dương
   - Mã số của nhân viên, kiểu chuỗi, được tạo bằng một chuỗi, trong chuỗi đó chứa thông tin số ngày làm việc kết hợp với họ tên
     - Ví dụ: nhân viên có tên là "Thanh", có số ngày làm việc là 30 vậy mã số của nhân viên được chương trình tạo sẽ là chuỗi "30Thanh"
   - Chức danh của nhân viên, kiểu chuỗi, chức danh của nhân viên cũng được chương trình tự tạo dựa trên thông tin số ngày làm việc, cụ thể như sau:

Số ngày làm việc	Chức danh
Từ 0 đến 365 ngày	Nhân viên
Từ 365 ngày đến 730 ngày	Quản lý
Từ 730 ngày đến 1460 ngày	Trưởng phòng
Trên 1460 ngày	Trưởng ban quản lý

   - Hệ số lương của nhân viên, kiểu số thực dương, hệ số lương của nhân viên cũng được chương trình tính dựa trên thông tin chức danh, cụ thể như sau:

Chức danh	Hệ số lương
Nhân viên	1.0
Quản lý	1.5
Trưởng phòng	2.25
Trưởng ban quản lý	4.0

a. Hãy viết các phương thức getter, setter cho toàn bộ thuộc tính cần thiết của lớp Nhân Viên.
b. Hãy viết các phương thức như sau:
   - Phương thức tạo lập với 5 tham số, ứng với 5 thông tin mô tả
   - Phương thức tạo lập với 2 tham số: họ tên và số ngày làm việc (các thông tin khác sinh viên tự xử lý theo yêu cầu mô tả, không sử dụng tham số mặc định)
   - Phương thức tạo lập sao chép, chỉ sao chép chức danh, hệ số lương và số ngày làm việc, ngoài ra các thông tin còn lại được tạo lập mặc định
   - Phương thức hủy Nhân Viên
c. Hãy viết phương thức để nhập và xuất thông tin lớp Nhân Viên

Bài tập tổng hợp cơ bản về lớp

2. Cài đặt các phương thức tạo lập (Constructor), tạo lập sao chép, hủy bỏ cho lớp Phân số đã làm trong bài thực hành trước.
3. Định nghĩa lớp DSPhanSo (danh sách phân số) dựa trên lớp Phân số ở câu 2 để lưu trữ và xử lý các thao tác trên mảng các phân số. Yêu cầu:
   a) Viết chương trình cho phép người dùng nhập vào danh sách các phân số
   b) Tính tổng các phân số
   c) Tìm phân số lớn nhất
   d) Tìm phân số nhỏ nhất

e) Sắp xếp danh sách phân số tăng dần
f) Sắp xếp danh sách phân số giảm dần

4. Định nghĩa lớp mảng một chiều (MangMotChieu) để lưu trữ danh sách các số nguyên và hỗ trợ xử lý các thao tác cơ bản trên mảng một chiều:
   a) Tạo mảng ngẫu nhiên
   b) Xuất mảng
   c) Đếm số lần xuất hiện của giá trị x trong mảng
   d) Kiểm tra dãy số nguyên có tăng dần hay không
   e) Tìm phần lẻ nhỏ nhất trong dãy số nguyên
   f) Tìm số nguyên tố lớn nhất

5. Xét đa thức theo biến x (đa thức một biến) bậc n có dạng như sau:

   $P(X)=a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} +...+ a_0$

   Trong đó: n là bậc của đa thức. $a_0, a_1, a_2,..., a_n$ là các hệ số tương ứng với từng bậc của đa thức.

   Định nghĩa lớp DaThuc biểu diễn khái niệm đa thức với các thao tác sau:
   a) Phương thức khởi tạo một đa thức có bậc bằng 0.
   b) Nhập đa thức
   c) Xuất đa thức
   d) Tính giá trị của đa thức khi biết giá trị của x
   e) Cộng hai đa thức
   f) Trừ hai đa thức

   Viết chương trình cho phép người dùng nhập vào hai đa thức, xuất các đa thức ra màn hình, tính tổng , hiệu hai đa thức và xuất kết quả ra màn hình.