

Bài thực hành 9

Mục tiêu

- Nạp chồng toán tử một ngôi
- Hàm thành viên
- Từ khóa friend và chuyển đổi kiểu tự động
- Tham chiếu

Bài mẫu

Đề bài:

Định nghĩa lớp Time biểu diễn đại lượng thời gian với 3 thành viên dữ liệu: hours là số giờ, minutes là số phút, seconds là số giây. Các phương thức chính của lớp:

- Phương thức kiến tạo không tham số đặt tất cả các thông số bằng 0.
- Phương thức chuẩn hóa standardize() biến đổi các thông số để số phút và số giây phải ở trong khoảng 0-59.
- Phương thức kiến tạo có tham số đặt giá trị ban đầu cho 3 thông số, sau đó chuẩn hóa bằng cách gọi tới standardize().
- Phương thức input() đọc 3 thông số vào từ bàn phím, sau đó chuẩn hóa bằng cách gọi tới standardize().
- Nạp chồng toán tử + bằng một hàm ngoài lớp Time. Hàm này có 2 tham số tham chiếu hằng kiểu Time, trả về một giá trị kiểu Time.

Lời giải:

```
// Developer: Diep Thi Hoang
// Description: Lop Thoigian bieu dien dai luong thoi gian theo gio, phut, giay
//             Minh hoa cach nap chong toan tu
#include <iostream>
using namespace std;

// Lop Thoigian bieu dien dai luong thoi gian theo gio, phut, giay
class Time{
public:
    Time();// Ham kien tao khong tham so
    Time(int h, int m, int s);// Ham kien tao co tham so, co gọi standardize()
    int getHours() const;// Lay so gio
    int getMinutes() const;// Lay so phut
    int getSeconds() const;// Lay so giay

    void setHours(int h);// Dat so gio
    void setMinutes(int m);// Dat so phut
    void setSeconds(int s);// Dat so giay

    void standardize();// Chuan hoa: minutes va seconds phai tu 0-59
    void output() const;// In ra man hinh cac thong so gio, phut, giay
    void input();// Nhap du lieu tu ban phim, co gọi standardize()

private:
    int hours;// gio
    int minutes;// phut
    int seconds;// giay
};
```

```
// Ham nap chong toan tu + cho lop Time
const Time operator+(const Time& t1, const Time& t2);

// -----
// Ham main
int main(){
    Time timeVar1(3, 70, 67);
    cout << "Bien Time thu nhat: ";
    timeVar1.output();
    cout << endl;

    cout << "Bien Time thu hai: ";
    Time timeVar2(0, 108, 53);
    timeVar2.output();
    cout << endl;

    cout << "Tong cua 2 bien Time: ";
    (timeVar1 + timeVar2).output();
    cout << endl;

    cout << "Nhap cac thong so cho bien Time thu ba:\n";
    Time timeVar3;
    timeVar3.input();
    cout << "Bien Time thu ba: ";
    timeVar3.output();
    cout << endl;

    cin.ignore(80, '\n');
    cout << "\nHay nhan ENTER de ket thuc chuong trinh...";
    cin.get();
    return 0;
}

// -----
// Dinh nghia cac ham

// Ham kien tao khong tham so
Time::Time(): hours(0), minutes(0), seconds(0){
}

// Ham kien tao co tham so, co goi standardize()
Time::Time(int h, int m, int s): hours(h), minutes(m), seconds(s){
    standardize();
}

// Chuan hoa: minutes va seconds phai tu 0-59
void Time::standardize(){
    int s = seconds;
    int m = minutes;
    int h = hours;

    seconds = s % 60;
    m += s / 60;
    minutes = m % 60;
    hours = h + m / 60;
}

// Lay so gio
int Time::getHours()const{
    return hours;
}

// Lay so phut
```

```

int Time::getMinutes()const{
    return minutes;
}

// Lay so giay
int Time::getSeconds()const{
    return seconds;
}

// Dat so gio
void Time::setHours(int h){
    hours = h;
}

// Dat so phut
void Time::setMinutes(int m){
    minutes = m;
}

// Dat so giay
void Time::setSeconds(int s){
    seconds = s;
}

// In ra man hinh cac thong so gio, phut, giay
void Time::output() const{
    cout << hours << " gio, " << minutes << " phut, " << seconds << " giay";
}

// Nhap du lieu tu ban phim, co goi standardize()
void Time::input(){
    cout << "Nhap so gio: ";
    cin >> hours;
    cout << "Nhap so phut: ";
    cin >> minutes;
    cout << "Nhap so giay: ";
    cin >> seconds;
    cin.ignore(80, '\n');
    standardize();
}

// Ham nap chong toan tu + cho lop Time
const Time operator+(const Time& t1, const Time& t2){
    int h = t1.getHours() + t2.getHours();
    int m = t1.getMinutes() + t2.getMinutes();
    int s = t1.getSeconds() + t2.getSeconds();
    return Time(h, m, s);
}

```

Kết quả

```

Bien Time thu nhat: 4 gio, 11 phut, 7 giay
Bien Time thu hai: 1 gio, 48 phut, 53 giay
Tong cua 2 bien Time: 6 gio, 0 phut, 0 giay
Nhap cac thong so cho bien Time thu ba:
Nhap so gio: 6
Nhap so phut: 3
Nhap so giay: 120
Bien Time thu ba: 6 gio, 5 phut, 0 giay

```

Bài tập

Câu 1. [time_outside.cpp]

- a) Xem xét định nghĩa hàm nạp chồng toán tử + của lớp Time trong Bài mẫu. Hãy giải thích vì sao ta nên có từ khóa const trước kiểu trả về.
- b) Hãy định nghĩa hàm nạp chồng toán tử - cho lớp Time: hàm này thực hiện phép trừ trên 2 tham số kiểu Time và trả về một giá trị Time. Viết chương trình đoạn chương trình chạy thử.
- c) Hãy định nghĩa hàm nạp chồng toán tử == cho lớp Time: hàm này thực hiện phép so sánh bằng trên 2 tham số kiểu Time và trả về một giá trị bool. Viết chương trình đoạn chương trình chạy thử.
- d) Hãy định nghĩa hàm nạp chồng toán tử - một ngôi cho lớp Time: hàm này trả về giá trị đảo dấu của 1 tham số kiểu Time. Viết chương trình đoạn chương trình chạy thử.
- e) Hãy định nghĩa hàm nạp chồng toán tử tự tăng trước ++ một ngôi cho lớp Time: hàm này tăng thông số seconds của tham số tham chiếu kiểu Time lên 1 đơn vị, sau đó gọi tới hàm standardize() để chuẩn hóa. Hàm trả về giá trị Time sau khi tăng. Viết đoạn chương trình chạy thử. Nhắc lại: nếu lớp Time nạp chồng toán tử tự tăng trước thì ta có thể gọi ++timeVar3.
- f) Hãy định nghĩa hàm nạp chồng toán tử tự tăng sau ++ một ngôi cho lớp Time: hàm này tăng thông số seconds của tham số tham chiếu kiểu Time lên 1 đơn vị, sau đó gọi tới hàm standardize() để chuẩn hóa. Hàm trả về giá trị Time trước khi tăng. Viết đoạn chương trình chạy thử. Nhắc lại: nếu lớp Time nạp chồng toán tử tự tăng sau thì ta có thể gọi timeVar3++.

Câu 2. [time_inside.cpp]

Trong câu 1, các toán tử nạp chồng cho lớp Time được khai báo ngoài lớp. Hãy khai báo và định nghĩa chúng dưới dạng phương thức thành viên của lớp.

Câu 3. [time_friend.cpp]

Chỉnh sửa lại định nghĩa và khai báo các hàm nạp chồng toán tử cho lớp Time trong câu 1 để sử dụng từ khóa friend.

Câu 4. [time_stream.cpp]

Tạo bản sao của mã nguồn câu 3 rồi bổ sung

- a) định nghĩa nạp chồng toán tử >> đọc từ luồng istream vào biến Time và
- b) định nghĩa nạp chồng toán tử << ghi thông số của biến Time ra luồng ra ostream.

Gợi ý

Câu 1. [time_outside.cpp]

Tham khảo Display 8.1 giáo trình Absolute C++.

Câu 2. [time_inside.cpp]

Tham khảo Display 8.2 giáo trình Absolute C++.

Câu 3. [time_friend.cpp]

Tham khảo Display 8.3 giáo trình Absolute C++.

Câu 4. [time_stream.cpp]

Tham khảo Display 8.5 giáo trình Absolute C++.