

Nội dung tuần 03

Luyện tập cách xây dựng các thành phần tĩnh của lớp đối tượng.

Hướng dẫn

Ví dụ 1

```
class SoPhuc
{
private:
    int thuc, ao;
    int sl;
    static int ssl;

public:
    SoPhuc();
    SoPhuc(const int &th, const int& a);
    SoPhuc(const SoPhuc& sp);
    ~SoPhuc();

    SoPhuc operator+(const SoPhuc& sp);
    int SoLuongSP();
    static int SSoLuongSP();
};

SoPhuc::SoPhuc()
{
    thuc = ao = 0;
    sl++;
    ssl++;
}

SoPhuc::SoPhuc(const int& th, const int& a)
{
    thuc = th;
    ao = a;
    sl++;
    ssl++;
}

SoPhuc::SoPhuc(const SoPhuc& sp)
{
    thuc = sp.thuc;
    ao = sp.a;
    sl++;
    ssl++;
}

SoPhuc::~~SoPhuc()
{
    sl--;
    ssl--;
}
```

```
SoPhuc SoPhuc::operator+(const SoPhuc& sp)
{
    SoPhuc kq(*this);
    kq.thuc += sp.thuc;
    kq.ao += sp.ao;
    return kq;
}

int SoPhuc::SoLuongSP()
{
    return sl;
}

int SoPhuc::SSoLuongSP()
{
    return ssl;
}

int SoPhuc::ssl = 0;
void main()
{
    SoPhuc sp1;
    SoPhuc sp2(5, 10);
    SoPhuc sp3(sp1);
    SoPhuc sp4 = sp2 + sp1;
    //cout << SoPhuc::SSoLuongSP() << endl;
    {
        SoPhuc sp5;
        SoPhuc sp6 = sp2 + sp4;
    }
    cout << sp4.SoLuongSP() << endl;           //???
    cout << SoPhuc::SSoLuongSP() << endl;      //4
}
```

- ✓ Mục tiêu là đếm số lượng đối tượng **SoPhuc** đã được tạo ra, nếu cài đặt theo thành phần của đối tượng như biến **sl** thì vấn đề là biến thực sự của đối tượng nào? (sp1, sp2 hay sp3,...) và khởi tạo biến như thế nào? Tóm lại là để đếm số lượng đối tượng đã tạo ra thì biến đếm không thuộc đối tượng cụ thể nào cả.
- ✓ Sử dụng biến là thành phần tĩnh (static) như **ssl** là đúng đắn.
- ✓ Vấn đề là vì sao kết quả là 4? (ngay theo khởi tạo thì thấy ít nhất 6 đối tượng đã được tạo ra). Chú ý dòng được highlight, tạo đó kết quả cũng bằng 4 (lý do là vì **sp5** và **sp6** được gom trong nhóm lệnh với cặp dấu {}, điều này nghĩa là **sp5** và **sp6** sẽ bị hủy ngay khi kết thúc nhóm lệnh).

Ví dụ 2

```
class Ngay
{
private:
    int d, m, y;
    static int arrNgayCuaThang[][13];
    static bool LaNamNhuan(const int& n);
};
```

```
public:
    Ngay();
    Ngay(const int& n);
    ~Ngay();
    friend ostream& operator<<(ostream& os, const Ngay& ng);
};

Ngay::Ngay()
{
    d = m = y = 1;
}

Ngay::Ngay(const int& n)
{
    d = (n >= 0) ? n : -n;
    m = y = 1;
    if (d == 0)
    {
        d = 1;
        return;
    }
    int idxN = LaNamNhuan(y);
    while(d > arrNgayCuaThang[idxN][m])
    {
        d -= arrNgayCuaThang[idxN][m];
        m++;
        if (m > 12)
        {
            m = 1;
            y++;
            idxN = LaNamNhuan(y);
        }
    }
}

Ngay::~~Ngay()
{
}

bool Ngay::LaNamNhuan(const int& n)
{
    return (n % 400 == 0) | (n % 4 == 0 & n % 100 != 0);
}

ostream& operator<<(ostream& os, const Ngay& ng)
{
    os << ng.d << "-" << ng.m << "-" << ng.y;
    return os;
}

int Ngay::arrNgayCuaThang[][13] = {{-1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31},
                                     {-1, 31, 29, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31}};

void main()
{
```

Hướng dẫn thực hành PP LT hướng đối tượng

```
    Ngay n1;  
    Ngay n2(1000);  
    Ngay n3(2000);  
    Ngay n4(3000);  
    Ngay n5(5000);  
    cout << n1 << endl << n2 << endl << n3 << endl;  
    cout << n4 << endl << n5 << endl;  
}
```

- ✓ Vận dụng thành phần tĩnh để cài đặt các trợ giúp cần thiết.
- ✓ Ở lớp đối tượng **Ngay** thì để dễ dàng tính toán số ngày tối đa tương ứng với tháng đồng thời giúp cho đoạn code gọn gàng dễ hiểu hơn thì nên sử dụng mảng lưu số ngày tối đa của tháng. Rõ ràng mảng này chỉ có ý nghĩa với lớp đối tượng **Ngay** và đồng thời cần ngăn ngừa mảng bị thay đổi không mong muốn thì nên khai báo mảng là thành phần tĩnh của lớp **Ngay**.
- ✓ Tương tự với hàm tiện ích kiểm tra 1 năm có phải là nhuận không.

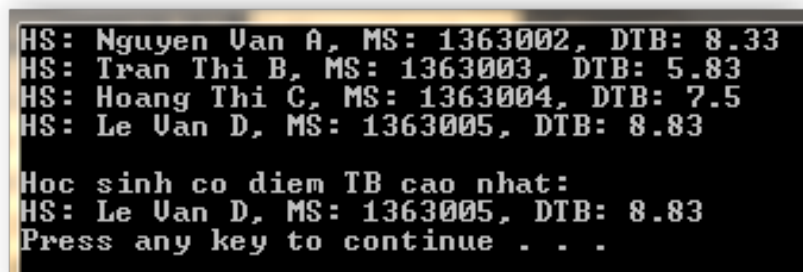
Bài tập

Bài 1

Khai báo và cài đặt lớp đối tượng **HocSinh** sao cho hàm main sau chạy đúng

```
void main()  
{  
    HocSinh hs1;  
    hs1.DatHoTen("Nguyen Van A");  
    hs1.GanDiem(7, 8, 10);  
    HocSinh hs2("Tran Thi B", 5, 8, 4.5);  
    HocSinh hs3("Hoang Thi C", -9.5, 8.5, 4.5);  
    HocSinh hs4("Le Van D", 7.5, 9, -10);  
    //"HS: Nguyen Van A, MS: 1363001, DTB: 7.53"  
    cout << hs1 << endl << hs2 << endl << hs3 << endl << hs4 << endl << endl;  
  
    cout << "Hoc sinh co diem TB cao nhat:" << endl;  
    cout << HocSinh::HSDiemTBCaoNhat() << endl;  
}
```

Biết rằng Mã số học sinh không được trùng và nhỏ nhất là 1363001.



```
HS: Nguyen Van A, MS: 1363002, DTB: 8.33  
HS: Tran Thi B, MS: 1363003, DTB: 5.83  
HS: Hoang Thi C, MS: 1363004, DTB: 7.5  
HS: Le Van D, MS: 1363005, DTB: 8.83  
  
Hoc sinh co diem TB cao nhat:  
HS: Le Van D, MS: 1363005, DTB: 8.83  
Press any key to continue . . .
```

Bài 2

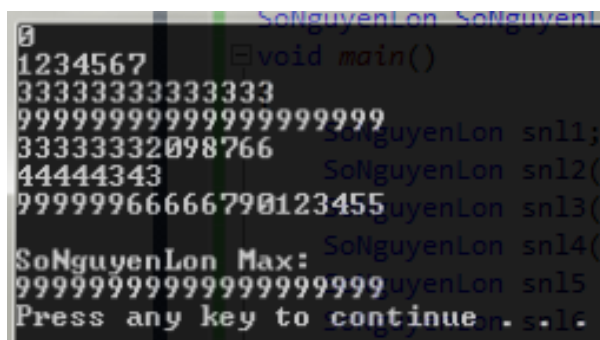
Khai báo và cài đặt lớp đối tượng SoNguyenLon sao cho hàm main sau chạy đúng

```
void main()
{
    SoNguyenLon snl1;
    SoNguyenLon snl2(1234567);
    SoNguyenLon snl3(3, 14);
    SoNguyenLon snl4(9, 20);
    SoNguyenLon snl5 = snl3 - snl2;
    SoNguyenLon snl6 = 45678910 - snl2;
    SoNguyenLon snl7 = snl4 - snl3 + 123456789;

    cout << snl1 << endl << snl2 << endl << snl3 << endl;
    cout << snl4 << endl << snl5 << endl << snl6 << endl;
    cout << snl7 << endl << endl;

    cout << "SoNguyenLon Max:" << endl;
    cout << SoNguyenLon::SNLMax() << endl;
}
```

Với kết quả như sau



The screenshot shows the output of the program in a terminal window. The output is as follows:

```
1234567
3333333333333333
9999999999999999
33333332098766
44444343
9999996666790123455
SoNguyenLon Max:
9999999999999999
Press any key to continue.
```

Biết rằng số chữ số tối đa của SoNguyenLon là 100 và không có giá trị âm (số bé hơn trừ số lớn hơn sẽ có kết quả là 0).