

CHƯƠNG 3

LỚP

- Lớp là khái niệm trung tâm của lập trình hướng đối tượng, trong ngôn ngữ C++, lớp là **kiểu dữ liệu do người dùng định nghĩa** và là sự mở rộng của kiểu cấu trúc trong ngôn ngữ C.
- Chương 3 trình bày: Định nghĩa lớp, khai báo đối tượng, truy cập đến các thành phần của lớp, con trỏ đối tượng, con trỏ **this**, hàm bạn, dữ liệu thành phần tĩnh, phương thức tĩnh, hàm tạo, hàm hủy.

3.1. Định nghĩa lớp

Cú pháp: Lớp được định nghĩa theo mẫu :
class tên_lớp

```
{  
    private: [Khai báo các thuộc tính]  
             [Định nghĩa các phương thức]  
    public : [Khai báo các thuộc tính]  
            [Định nghĩa các phương thức]  
    protected : [Khai báo các thuộc tính]  
                [Định nghĩa các phương thức]  
};
```

Thành phần của lớp được tổ chức thành các vùng:

- **Vùng riêng** (private): Các thành phần private chỉ được sử dụng bên trong lớp.
- **Vùng được bảo vệ** (protected): Các thành phần protected được sử dụng bên trong lớp và trong lớp dẫn xuất của lớp đó.
- Đối tượng của lớp **KHÔNG** truy cập được vào các vùng private và protected.
- **Vùng chung** (public): Các thành phần public được phép sử dụng ở cả bên trong và bên ngoài lớp. Đối tượng của lớp **ĐƯỢC PHÉP** truy cập vào vùng này.

- **Khai báo các thuộc tính của lớp:** được thực hiện giống như việc khai báo biến.
- **Định nghĩa các phương thức:** Các phương thức có thể được xây dựng bên ngoài hoặc bên trong định nghĩa lớp.

Các phương thức khi định nghĩa ở bên ngoài lớp thực hiện theo cú pháp sau đây:

Kiểu Tên_lớp::Tên phương thức(khai báo các tham số)

{ ... }

và bên trong lớp chỉ ghi nguyên mẫu phương thức.

Ví dụ 3.1. Định nghĩa lớp Sinhvien để mô tả các đối tượng sinh viên, gồm các thuộc tính: Mã sinh viên, họ tên, điểm trung bình. Các phương thức: nhập xuất dữ liệu cho đối tượng sinh viên.

```
//sinhvien0.cpp
```

```
class Sinhvien
```

```
{
```

```
    private:
```

```
        char masv[10], ht[30];
```

```
        float dtb;
```

```
    public:
```

```
        void nhap(); //nhap du lieu cho doi tuong gọi  
                //   đến hàm nhap
```

```
        void xuat(); //xuat du lieu của doi tuong gọi  
                //   đến hàm xuat
```

```
};
```

```
void Sinhvien::nhap()
```

```
{  
    cout<<"\n Nhap ma sv :"; cin.get(masv,10);  
    cin.ignore();  
    cout<<"\n Nhap ho ten sv :";  
    cin.get(ht,30);  
    cout<<"\n Diem trung binh :";  
    cin>>dtb;  
    cin.ignore();  
}
```

```
void Sinhvien::xuat()
```

```
{  
    cout<<setw(10)<<masv  
        <<setw(30)<<ht  
        <<setw(5)<<dtb<<endl; }
```

Nhận xét:

1. Các phương thức `nhap()`, `xuat()` có thể sử dụng các thuộc tính của lời là `masv`, `ht`, `dtb` và không khai báo lại các thuộc tính này.
2. Các phương thức `nhap()`, `xuat()` tác động lên 1 đối tượng gọi đến phương thức.

Chú ý: Các hàm nhập(), xuất() có thể xây dựng bên trong lớp ?

3.2. Khai báo đối tượng, mảng đối tượng

3.2.1. Khai báo đối tượng

Sau khi định nghĩa lớp, ta có thể khai báo các biến thuộc kiểu lớp. Các biến này được gọi là các đối tượng.

Cú pháp khai báo biến đối tượng như sau:

Tên lớp Danh sách biến ;

Mỗi đối tượng sau khi khai báo sẽ được cấp phát một vùng nhớ riêng để chứa các thuộc tính của chúng. Không có vùng nhớ riêng để chứa các phương thức cho mỗi đối tượng. Các phương thức sẽ được sử dụng chung cho tất cả các đối tượng cùng lớp.

Để truy cập đến thuộc tính và phương thức của lớp từ đối tượng, ta dùng cú pháp:

Tên đối tượng. Tên thuộc tính

Tên đối tượng. Tên phương thức(...)

Ví dụ: Hàm main() minh họa việc nhập xuất dữ liệu cho 2 đối tượng của lớp Sinhvien

```
int main()  
{  
    Sinhvien sv1, sv2;  
  
    sv1.nhap(); //nhap du lieu cho sv1  
    sv2.nhap(); //nhap du lieu cho sv2  
  
    sv1.xuat(); //xuat du lieu cua sv1  
    sv2.xuat(); //xuat du lieu cua sv2  
  
    return 0;  
}
```

//cho biet loi cua chuong trinh sau

```
#include <iostream>
#include <iomanip>
using namespace std;
```

```
class A
{
    private:
        int a;
    public:
        float b;
```

```
};
```

```
int main()
```

```
{
    A ob;
    ob.a = 5;
    ob.b = 6.8;
    return 0;
}
```

**Thực hiện: gán dữ liệu
a, b cho đối tượng ob ?**

3.2.2. Khai báo mảng đối tượng

Việc khai báo mảng đối tượng được thực hiện theo cú pháp sau:

Tên lớp Danh sách mảng đối tượng ;
Để truy cập đến thuộc tính và phương thức của
lớp từ phần tử mảng đối tượng, ta dùng cú pháp:
 phần tử mảng. Tên thuộc tính
 phần tử mảng. Tên phương thức(...)

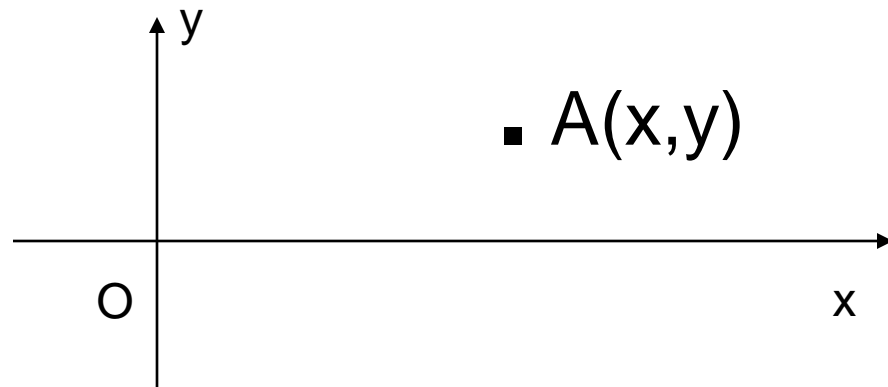
Ví dụ: Khai báo và nhập dữ liệu cho không quá 50
đối tượng của lớp Sinhvien và xuất ra màn hình:

```
int main()
{ Sinhvien  sv[50];
  int i,n;
  cout<<"\n Nhap so sinh vien :";
  cin>>n; cin.ignore();
  for (i = 0;i<n;i++)
    sv[i].nhap(); //Nhập dữ liệu cho sv thứ i
  cout<<"\nDanh sach sinh vien ";
  for (i=0; i<n ; i++)
    sv[i].xuat();//xuất dữ liệu của sv thứ i
  return 0;
}
```

Thảo luận:

1. Định nghĩa lớp mô tả các đối tượng là sách, gồm các thuộc tính: mã sách, tên sách, tác giả, số lượng, nhà xuất bản. Các phương thức Nhập, Xuất dữ liệu của đối tượng. Viết Chương trình minh họa trên danh sách không quá 50 đối tượng sách.
2. Sử dụng lớp Sinhvien, hãy viết chương trình in danh sách sinh viên có **điểm trung bình** lớn nhất.

3. Định nghĩa lớp để mô tả các đối tượng là điểm trong mặt phẳng tọa độ Oxy, đặc trưng bởi hoành độ và tung độ. Các phương thức nhập, xuất dữ liệu và minh họa chương trình trên một mảng không quá 50 đối tượng, in ra danh sách các điểm thuộc trục hoành.



3.3. Truy cập tới các thành phần của lớp

- Đối với đối tượng thông thường:
Tên đối tượng. Tên thuộc tính
Tên đối tượng. Tên phương thức(...)
- Đối với phần tử mảng đối tượng:
phần tử mảng. Tên thuộc tính
phần tử mảng. Tên phương thức(...)
- Đối với con trỏ đối tượng:
tên con trỏ -> Tên thuộc tính
tên con trỏ -> Tên phương thức(...)