

# **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

# Lập trình hướng đối tượng

Bài toán: Cần làm 1 phần mềm quản lý học sinh. Số lượng người thực hiện dự án là 10 người.

Nếu chỉ làm trên file main, quá nhiều người thao tác sẽ dẫn đến xung đột code với nhau. Ngoài ra file main quá nhiều sẽ dẫn đến thuật toán không tối ưu, rõ ràng.

# Định nghĩa Class(Lớp)

Được định nghĩa như một kiểu dữ liệu đơn giản hay một cấu trúc dữ liệu, được khai báo bằng từ khóa **class**, nó chứa các biến (còn gọi là thuộc tính - *property*) hay các hàm (còn gọi là phương thức - *method*).

```
class [TÊN CLASS] {
```

```
    //Khai báo các phương thức và thuộc tính
```

```
};
```

# Định nghĩa Class(Lớp)

Các phần tử của **class** được quản lý bởi ba thuộc tính truy cập:

1. *private*,
2. *protected*
3. *public*

Các phần tử *private* không thể được truy cập bên ngoài **class** mà chỉ có thể được truy cập thông qua các phương thức của **class** chứa chúng. Ngược lại, các phần tử *public* có thể được truy cập ở bất kỳ **class** nào.

Các phần tử *protected* tương tự như *private* tuy nhiên khác nhau ở kế thừa. Sẽ được nói ở các chương sau.

# Định nghĩa Class(Lớp)

```
class Box {  
    private:  
        double length;  
        double width;  
        double height;  
    public:  
        double area() {  
            return width * height;  
        }  
        double volume() {  
            return width * height * length  
        }  
};
```

# Định nghĩa Class(Lớp)

```
class Box {  
    private:  
        double length;  
        double width;  
        double height;  
    public:  
        double area();  
        double volume();  
};  
double Box::area() {  
    return width * height;  
}  
double Box::volume() {  
    return width * height * length;  
}
```

# Định nghĩa Object(Đối tượng)

Đối tượng được tạo ra từ 1 Lớp. Nó là biểu diễn của 1 Lớp. 1 Lớp muốn sử dụng được phải thông qua đối tượng.

**[TÊN CLASS] [TÊN CÁC BIẾN];**

**[TÊN CLASS]\* [TÊN BIẾN] = new [TÊN CLASS]();**

# Định nghĩa Object(Đối tượng)

```
class Box {  
    private:  
        double length, width, height;  
    public:  
        void set(double l, double w, double h);  
        double volume();  
};  
void Box::set(double l, double w, double h) {  
    length = l;  
    width = w;  
    height = h;  
}  
double Box::volume() {  
    return width * height * length;  
}
```



# Định nghĩa Object(Đối tượng)

```
Box b1, b2, b3;  
Box *box1 = new Box();  
Box boxes[100];  
Box* b = new Box[100];  
  
b1.set(1, 2, 3);  
cout<<b1.volume(); //6  
  
box1->set(2, 4, 6);  
cout<<box1->volume(); //48  
  
boxes[0].set(10, 20, 30);  
cout<< boxes[1].volume(); //0
```

# Định nghĩa Object(Đối tượng)

```
Box * box2;  
box2 = new Box();  
  
box2->set(2, 3, 4);  
cout<<box2->volume(); //24  
  
box2 = new Box()  
  
box2 ->set(0, 2, 3);  
cout<<box2->volume(); //0
```

# Định nghĩa Object(Đối tượng)

```
Box box4, box3;
```

```
box3.set(2, 3, 4);
```

```
box4.set(4, 5, 6);
```

```
cout<<box3.volume(); //24
```

```
cout<<box4.volume(); //120
```

```
box3 = box4;
```

```
cout<<box3.volume(); //120
```

```
box4.set(1, 2, 3);
```

```
cout<<box3.volume(); //120
```

# Con trỏ this

Con trỏ **this** dùng để phân biệt các biến trong 1 *phương thức(hàm)* và các *thuộc tính*. Trong 1 phương thức, có 1 biến tên trùng 1 thuộc tính, nếu không sử dụng con trỏ *this*, biến được ưu tiên sẽ là biến của phương thức.

```
void Box::set(double length, double width, double height) {  
    this->length = length;  
    this->width = width;  
    this->height = height;  
}
```

# Con trỏ this

```
void Box::set(double length, double width, double height) {  
    length = length;  
    width  = width;  
    height = height;  
}
```

```
void Box::set(double length, double width, double height) {  
    length = 10;  
    width  = 20;  
    height = 30;  
}
```

# Con trỏ this

```
void Box::set(double length, double width, double height) {  
    this->length    = length;  
    this->width     = width;  
    this->height    = height;  
}
```

```
void Box::set(double length, double width, double height) {  
    this->length    = 10;  
    this->width     = 20;  
    this->height    = 30;  
}
```

# Từ khoá static

```
class Box {  
    public:  
        static double length;  
        void set(double l, double w, double h);  
        double volume();  
};  
static double Box::length = 5;  
  
//Hàm main  
Box b, c;  
b.length = 10;  
cout<<c.length; // 10
```

# Từ khoá static

```
class Box {  
    public:  
        static double length;  
        void set(double l, double w, double h);  
        double volume();  
};  
static double Box::length = 5;  
  
//Hàm main  
Box b, c;  
Box::length = 20;  
cout<<c.length; // 20
```



# Từ khoá static

```
class Box {  
    public:  
        static double length;  
        void set(double l, double w, double h);  
        double volume();  
};  
static double Box::length = 5;  
  
//Hàm main  
Box b, c;  
b.length = 30;  
cout<<Box::length; // 30
```

# Quản lý tập tin

Ta nên tách các *class* thành các file .h và .cpp riêng biệt để cho chương trình gọn hơn.

File .h sẽ chứa các mô tả của *class*.

File .cpp sẽ chứa các chi tiết của các phương thức trong *class*.

# Quản lý tập tin

```
//File Box.h
class Box {
    private:
        double length, width, height;
    public:
        void set(double l, double w, double h);
        double volume();
};
```

# Quản lý tập tin

```
//File Box.cpp
void Box::set(double l, double w, double h) {
    length = l;
    width = w;
    height = h;
}
double Box::volume() {
    return width * height * length;
}
```

# Quản lý tập tin

```
//File main.cpp
#include <iostream>
#include "Box.h"
using namespace std;

int main() {
    Box* box3;
    box3->set(2, 3, 4);
    double value = box3->volume();
    cout<<value;
    return 1;
}
```