LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Chương 3: Lớp và đối tượng (p1)

Trình bày: ThS. Lê Thanh Trọng

NỘI DUNG

1. Giới thiệu
2. Khai báo lớp
3. Các thành phần của lớp
4. Cơ chế tạo lập các lớp
5. Định nghĩa hàm thành phần
6. Tạo lập đối tượng

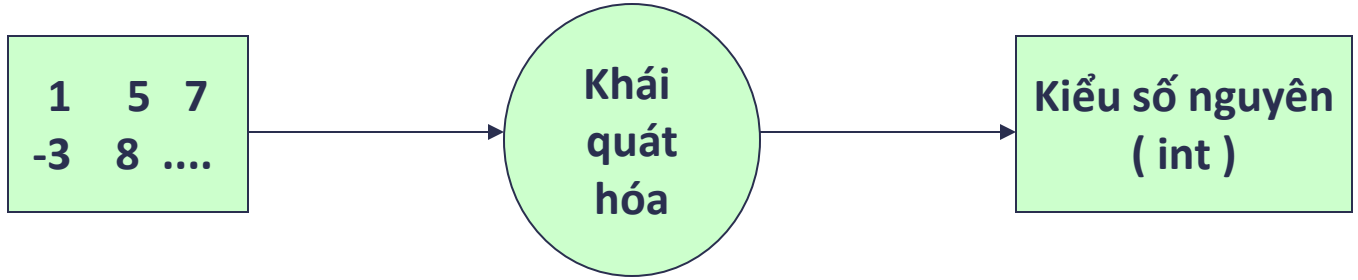
NỘI DUNG

Giới thiệu

- Một lớp bao gồm:
  - Các thành phần dữ liệu (thuộc tính)
  - Các hàm thành phần (phương thức)
- Lớp trong C++ thực chất là một kiểu dữ liệu do người sử dụng định nghĩa
- Trong C++, dùng từ khóa class để chỉ điểm bắt đầu của một lớp sẽ được cài đặt

Lớp đối tượng - class

- Lớp là một mô tả trừu tượng của nhóm các đối tượng cùng bản chất, ngược lại mỗi một đối tượng là một thể hiện cụ thể cho những mô tả trừu tượng đó
- Lớp là cái được thiết kế và lập trình
- Đối tượng là cái được tạo (từ một lớp) tại thời gian chạy



NỘI DUNG

1. Giới thiệu
2. **Khai báo lớp**
3. Các thành phần của lớp
4. Cơ chế tạo lập các lớp
5. Định nghĩa hàm thành phần
6. Tạo lập đối tượng
7. Phạm vi truy xuất

Khai báo lớp

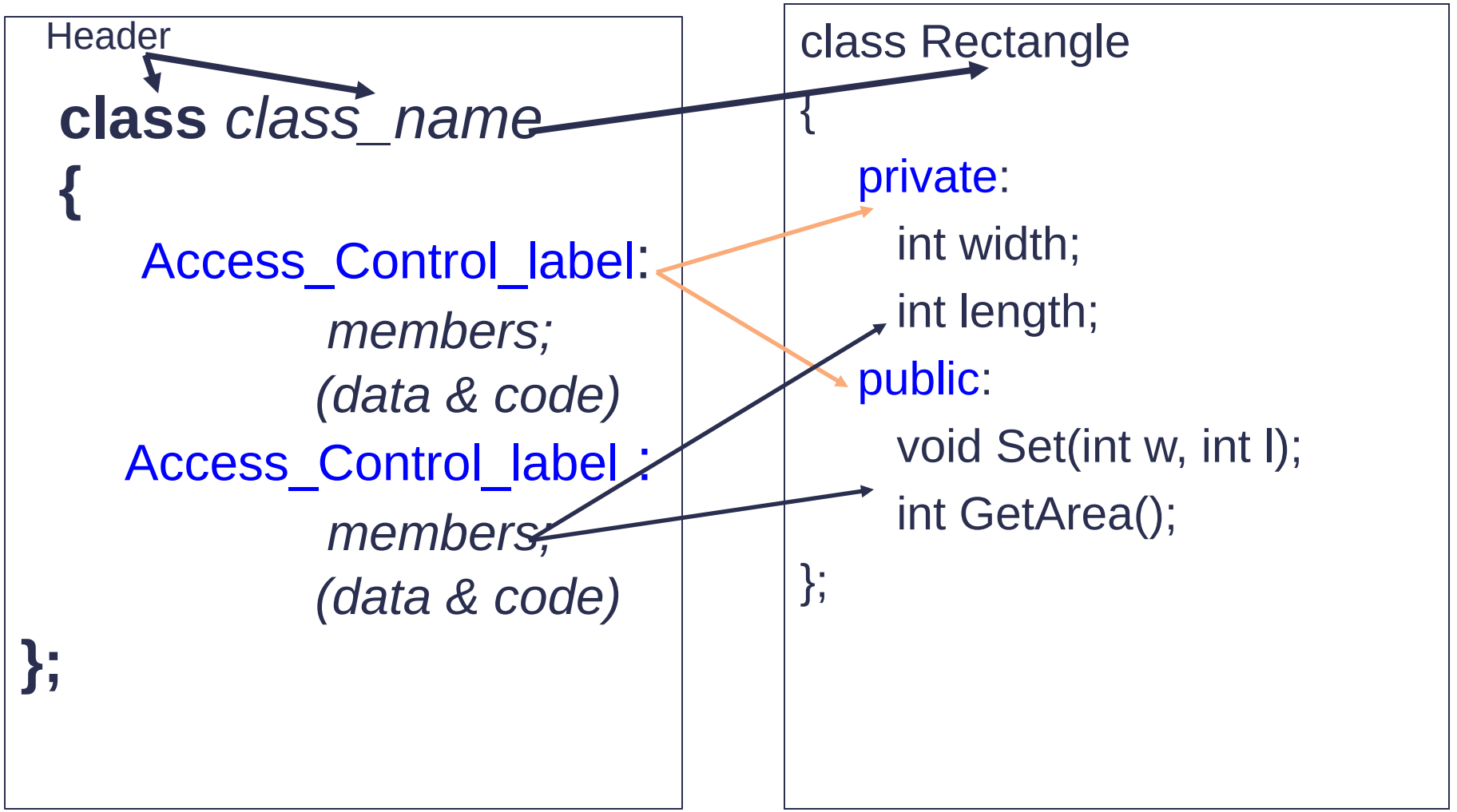
```
class <TenLop>
{
        //Thành phần dữ liệu (thuộc tính)

        //Thành phần xử lý (phương thức)
};
```

Khai báo lớp

```
class <tên_lớp> {
  private:
   <khai báo thành phần riêng trong từng đối tượng>
  protected:
   <khai báo thành phần riêng trong từng đối tượng, có thể truy cập từ lớp dẫn xuất >
  public:
    <khai báo thành phần công cộng>
};
```

Khai báo lớp


Header
class class_name
{
Access_Control_label:
members;
(data & code)
Access_Control_label :
members;
(data & code)
};
class Rectangle
{
private:
int width;
int length;
public:
void Set(int w, int l);
int GetArea();
};

NỘI DUNG

1. Giới thiệu
2. Khai báo lớp
3. **Các thành phần của lớp**
4. Cơ chế tạo lập các lớp
5. Định nghĩa hàm thành phần
6. Tạo lập đối tượng

Các thành phần của lớp

- Thuộc tính: Các thuộc tính được khai báo giống như khai báo biến trong C
- Phương thức: Các phương thức được khai báo giống như khai báo hàm trong C. Có hai cách định nghĩa thi hành của một phương thức
  - Định nghĩa thi hành trong lớp
  - Định nghĩa thi hành ngoài lớp

Ví dụ thành phần của lớp

Lớp	Thuộc tính	Phương thức
Xe máy	- Nhãn hiệu - Ngày sản xuất - Trọng lượng	- Khởi động - Chạy - Dừng
Sinh viên	- Tên sinh viên - Mã sinh viên - Điểm trung bình	- Nhập thông tin - Xuất thông tin - Xếp loại
Phân số	- Tử số - Mẫu số	- Nhập - Xuất - Tính tổng

NỘI DUNG

1. Giới thiệu
2. Khai báo lớp
3. Các thành phần của lớp
4. **Cơ chế tạo lập các lớp**
5. Định nghĩa hàm thành phần
6. Tạo lập đối tượng

Cơ chế tạo lập các lớp

- Xác định các thuộc tính (dữ liệu)
  - Những gì mà ta biết về đối tượng – giống như một struct
- Xác định các phương thức (hình vi)
  - Những gì mà đối tượng có thể làm
- Xác định các phạm vi truy cập phù hợp (được đề cập phần sau)

NỘI DUNG

1. Giới thiệu
2. Khai báo lớp
3. Các thành phần của lớp
4. Cơ chế tạo lập các lớp
5. **Định nghĩa hàm thành phần**
6. Tạo lập đối tượng

Định nghĩa hàm thành phần

❖Định nghĩa các hàm thành phần ở bên ngoài khai báo lớp:

<tên kiểu giá trị trả về> <tên lớp>::<tên hàm> (<danh sách tham số>)

{

<nội dung >

}

Ví dụ:

```
void Point::Xuat() {
    //……..
}
```

Định nghĩa hàm thành phần

```
class Rectangle{
    private:
       int width, length;
    public:
       void set (int w, int l);
       int area() { return width*length; }
};
```

Rectangle.h

Tên lớp

Tên phương thức

Rectangle.cpp

inline

```
void Rectangle :: set (int w, int l)
{
     width = w;
     length = l;
}
```

Toán tử phạm vi

main.cpp

```
r1.set(5,8);

rp->set(8,10);
```

NỘI DUNG

1. Giới thiệu
2. Khai báo lớp
3. Các thành phần của lớp
4. Cơ chế tạo lập các lớp
5. Định nghĩa hàm thành phần
6. **Tạo lập đối tượng**

Tạo lập đối tượng

- Khai báo và tạo đối tượng:

  <tên lớp> <tên đối tượng>;

- Gọi hàm thành phần của lớp

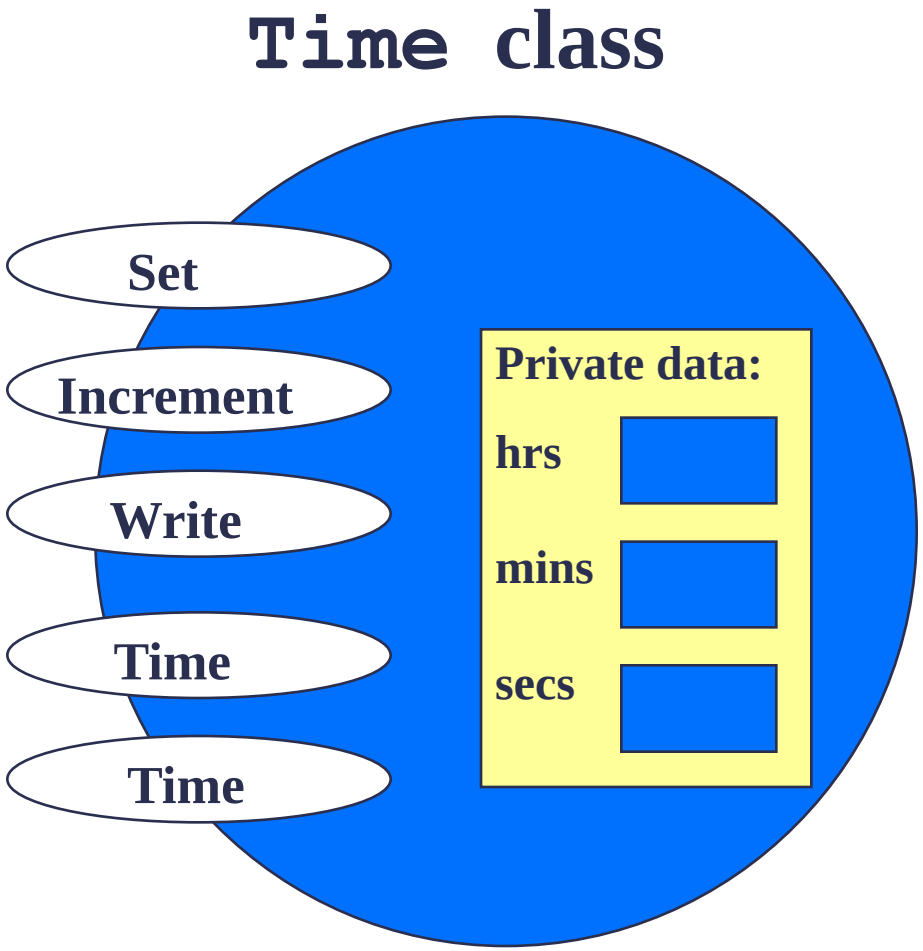
  <tên đối tượng>.<tên hàm thành phần> (<danh sách các tham số nếu có>);

  <tên con trỏ đối tượng>→<tên hàm thành phần> (<danh sách các tham số nếu có>);

Ví dụ: Class Time

```
class  Time {
    public:
            void    Set (int  hours , int  minutes , int  seconds);
            void     Increment ( );
            void     Write ( )  const;
            Time (int  initHrs, int  initMins,  int  initSecs );      //constructor
            Time ( );                                                 //default constructor
   private:
            int          hrs;
            int          mins;
            int          secs;
} ;
```

Sơ đồ mô tả lớp Time



Khai báo đối tượng

```
class Rectangle
{
    private:
        int width;
        int length;
    public:
        void set(int w, int l);
        int area();
};
```

r1 được cấp phát bộ nhớ

```
main()
{
    Rectangle r1;
⇨r1.set(5, 8);
}
```

r1

width = 5
length = 8

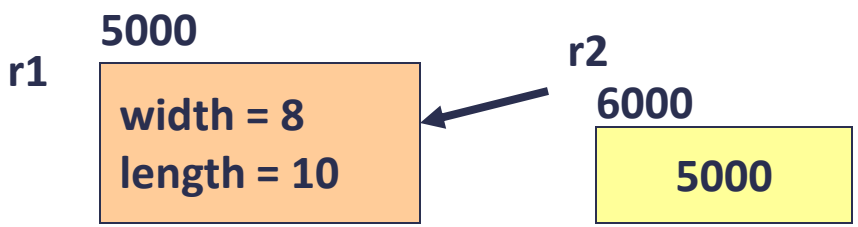
Khai báo đối tượng

```
class Rectangle
{
   private:
      int width;
      int length;
   public:
      void set(int w, int l);
      int area();
};
```

r2 là con trỏ chỉ đến đối tượng Rectangle

```
main()
{
    Rectangle r1;
    r1.set(5, 8);

    Rectangle *r2;
    r2 = &r1;
    r2->set(8,10);
}
```



Khai báo đối tượng

```
class Rectangle
{
    private:
        int width;
        int length;
    public:
        void set(int w, int l);
        int area();
};
```

r3 được cấp phát bộ nhớ

```
main()
{
    Rectangle *r3;
    r3 = new Rectangle();

    r3->set(80,100);

    delete r3;
    r3 = NULL;
}
```

r3

6000

NULL

Ví dụ

- Xây dựng lớp Điểm (Point) trong hình học 2D
  - Thuộc tính
    - Tung độ (float)
    - Hoành độ (float)
  - Thao tác (phương thức)
    - Nhập, Xuất
    - Di chuyển theo vector (m,n)
    - Tính khoảng cách với một điểm

Viết chương trình nhập vào 2 điểm, xuất ra 2 điểm vùa nhập, khoảng cách giữa 2 điểm đó và tọa độ mới của 2 điểm khi tịnh tiến theo vector có giá trị do người dùng nhập vào.

Ví dụ

```
//*Point.h
#include <iostream.h>
using namespace std;
class Point {
        /*khai báo các thành phần dữ  liệu riêng*/
        private:
                int x, y;
         /*khai báo các hàm thành phần công cộng*/
        public:
                void KhoiTao(int ox, int oy);
                void DiChuyen(int dx, int dy);
                void Xuat();
   };
```

Ví dụ

```
//Point.cpp
void Point::KhoiTao(int ox, int oy) {
    cout<<"Ham thanh phan khoi tao gia tri\n";
    x = ox; y = oy;
    //x, y là các thành phần của đối tượng gọi hàm thành phần
 }
void Point::DiChuyen(int dx, int dy) {
    cout<<"Ham thanh phan di chuyen\n";
     x += dx; y += dy;
 }
void Point::Xuat() {
    cout<<"Ham thanh phan xuat\n";
    cout<<"Toa do: "<<x<<","<<y<<"\n";
 }
```

Ví dụ

```
void main() {
    Point p;
    p.KhoiTao(2,4);
    p.Xuat();
    p.DiChuyen(1,2);
    p.Xuat();
}
```

```
Ham thanh phan khoi tao
Ham thanh phan xuat
Toa do: 2,4
Ham thanh phan di chuyen
Ham thanh phan xuat
Toa do: 3,6
```

Tóm tắt bài học lớp và đối tượng (p1)

- Lớp là một mô tả trừu tượng của nhóm các đối tượng cùng bản chất, ngược lại mỗi một đối tượng là một thể hiện cụ thể cho những mô tả trừu tượng đó
- Một lớp bao gồm:
  - Các thành phần dữ liệu (thuộc tính)
  - Các hàm thành phần (phương thức)
- Lớp trong C++ thực chất là một kiểu dữ liệu do người sử dụng định nghĩa
- Trong C++, dùng từ khóa class để chỉ điểm bắt đầu của một lớp sẽ được cài đặt

Tóm tắt bài học lớp và đối tượng (p1)

❖Cú pháp khai báo lớp

```
class <tên_lớp> {
  private:
   <khai báo thành phần riêng trong từng đối tượng>
  protected:
   <khai báo thành phần riêng trong từng đối tượng, có thể truy cập từ lớp dẫn xuất >
  public:
    <khai báo thành phần công cộng>
};
```

Tóm tắt bài học lớp và đối tượng (p1)

- Cơ chế tạo lập lớp
  - Xác định các thuộc tính (dữ liệu): Những gì mà ta biết về đối tượng – giống như một struct
  - Xác định các phương thức (hình vi): Những gì mà đối tượng có thể làm
  - Xác định các phạm vi truy cập phù hợp