

# Thành phần tĩnh, ba vấn đề con trỏ và mẫu Singleton

---

Nguyễn Khắc Huy

BMCNPM – ĐHKHTN TPHCM  
10/2015



# Nội dung

- Thành phần tĩnh
- Ba vấn đề con trỏ
- Mẫu thiết kế Singleton

cuu duong than cong . com

cuu duong than cong . com



# Thành phần tĩnh

## □ Thành phần của lớp (class members):

### ✓ Thành phần đối tượng (instance members).

- Thuộc tính và phương thức thông thường.
- Mỗi đối tượng có bản sao riêng.

### ✓ Thành phần tĩnh (static members).

- Thuộc tính và phương thức tĩnh.
- Các đối tượng dùng chung.

**Thành phần dùng chung cho  
MỌI đối tượng của lớp!!**



p1: PhanSo

- Tử số
- Mẫu số

p2: PhanSo

- Tử số
- Mẫu số

# Thành phần tĩnh

- Khai báo và sử dụng:
  - ✓ Dùng từ khóa static.
  - ✓ Truy xuất bằng toán tử ::.

```
class PhanSo
{
private:
    static int m_iGiaTriLN;
public:
    static int layGiaTriLN();
private:
    int m_iTu;
    int m_iMau;
};
```

```
PhanSo::m_iGiaTriLN = 10000;
```

```
void main()
```

```
{
```

```
    PhanSo p1(1, 2);
```

```
    PhanSo p2(2, 3);
```

```
    int x1 = PhanSo::layGiaTriLN();
```

```
    int x2 = p1.layGiaTriLN();
```

```
}
```



# Nội dung

- Thành phần tĩnh
- Ba vấn đề con trỏ
- Mẫu thiết kế Singleton

cuu duong than cong . com

cuu duong than cong . com



# Ví dụ

```
class IntArray
```

```
{
```

```
private:
```

```
    int    m_iSize;
```

```
    int    *m_pData;
```

```
public:
```

```
    IntArray(int iSize);
```

```
};
```

```
IntArray :: IntArray(int iSize)
```

```
{
```

```
    m_iSize = iSize;
```

```
    m_pData = new int[m_iSize];
```

```
}
```

```
void main()
```

```
{
```

```
    IntArray a1(5);
```

```
    IntArray a2(a1);
```

```
    a2 = a1;
```

```
}
```



# Nội dung

- Thành phần tĩnh
- Ba vấn đề con trỏ
- Mẫu thiết kế Singleton

cuu duong than cong . com

cuu duong than cong . com



# Ví dụ

```
class GlobalClass
```

```
{
```

```
    private:
```

```
        int value;
```

cuu duong than cong . com

```
    public:
```

```
        GlobalClass(int v = 0) {    value = v;    }
```

```
        int getValue() {    return value;    }
```

cuu duong than cong . com

```
        void setValue( int v ) {    value = v;    }
```

```
};
```





# Ví dụ (tt.)

```
void foo()
```

```
{  
    if (!gPtr) // Initialization  
        gPtr = new GlobalClass;  
    gPtr->setValue(1);  
    cout << "\nfoo:" << gPtr->getValue();  
}
```

```
void bar()
```

```
{  
    if (!gPtr)  
        gPtr = new GlobalClass;  
    gPtr ->setValue(2);  
    cout << "\nbar:" << gPtr ->getValue();  
}
```

```
// Default initialization  
GlobalClass *gPtr = 0;
```

```
int main()
```

```
{  
    if (! gPtr)  
        gPtr = new GlobalClass;  
  
    cout << "\nmain:" << gPtr->getValue();  
    foo();  
    bar();  
}
```

**Có cách nào  
tốt hơn?**

# Vấn đề



cuu duong than cong . com

cuu duong than cong . com



# Mẫu Singleton

```
class GlobalClass
```

```
{  
    int value;  
    static GlobalClass * instance;  
    GlobalClass(int v = 0) { value = v; }  
public:
```

```
public:
```

cuu duong than cong . com

```
    int getValue() { return value; }  
    void setValue( int v ) { value = v; }  
    static GlobalClass *instance() {  
        if (!instance)
```

cuu duong than cong . com

```
            instance = new GlobalClass;  
        return instance;  
    }  
};
```



# Ví dụ (tt.)

```
void foo() {  
    GlobalClass::instance()->setValue(1);  
    cout << "\nfoo:" << GlobalClass::instance() ->getValue() ;  
}
```

```
void bar() {  
    GlobalClass::instance()->setValue(2);  
    cout << "\nbar:" << GlobalClass::instance() ->getValue();  
}
```

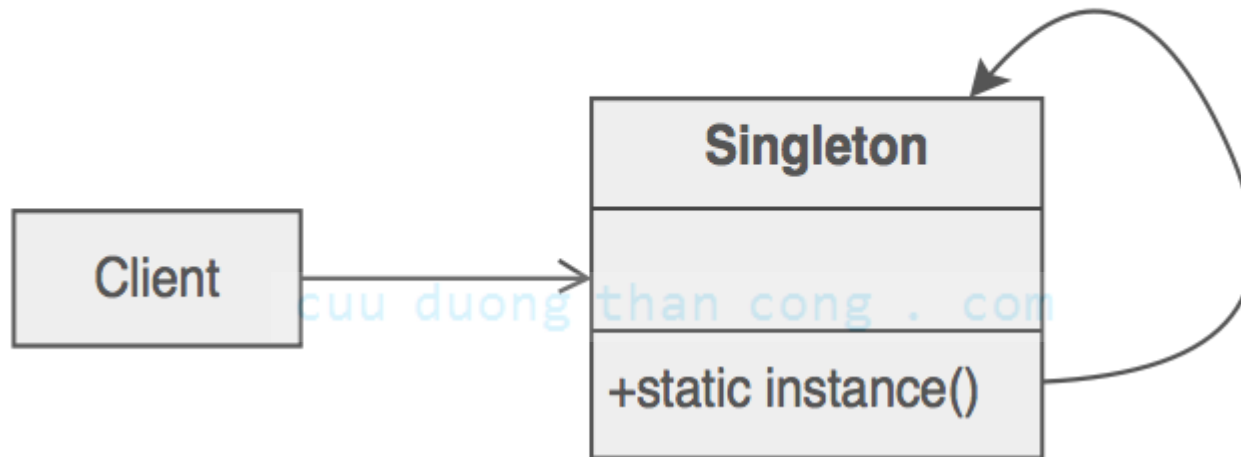
```
GlobalClass *GlobalClass::instance =0; // Default initialization
```

```
int main() {  
    cout << "\nmain:" << GlobalClass::instance() ->getValue();  
    foo();  
    bar();  
}
```



# Mẫu thiết kế Singleton

- Lớp đối tượng chỉ cần có duy nhất một thể hiện
- Ba vấn đề con trỏ trong Singleton



# Mẫu thiết kế Singleton

- Định nghĩa một thuộc tính tĩnh là một thể hiện đối tượng của chính lớp đối tượng.
- Định nghĩa một phương thức tĩnh công khai truy xuất thể hiện trên.
- Khởi tạo ở lần đầu truy xuất cho thể hiện trong phương thức truy xuất công khai.
- Tất cả các hàm dựng được khai báo ở tầm vực protected hoặc private.
- Bên ngoài chỉ được phép sử dụng đối tượng thông qua thể hiện cung cấp bởi phương thức truy xuất công khai.



# Bài tập

**Bài tập 5.1:** Xây dựng các lớp cần thiết để thực hiện các thao tác trên tam giác:

- Tính chu vi và diện tích.
- Tìm trọng tâm (giao 3 đường trung tuyến).
- Tìm tâm đường tròn ngoại tiếp (giao 3 đường trung trực).

(Gợi ý)

- Xây dựng 3 lớp điểm, đường thẳng, tam giác.
- Lớp điểm biểu diễn đỉnh tam giác.
- Lớp đường thẳng có các phương thức:
  - Khởi tạo đường thẳng đi qua 2 điểm.
  - Tìm giao điểm với đường thẳng khác.
  - Tìm đường thẳng vuông góc đi qua một điểm.





# Lời cảm ơn

- Nội dung được xây dựng dựa trên slide trình bày của Thầy Đinh Bá Tiến, Thầy Nguyễn Minh Huy.

cuu duong than cong . com

cuu duong than cong . com

