静态成员就是在成员变量和成员函数前加上关键字static，称为静态成员

静态成员分为：

- 静态成员变量
  - **所有对象共享同一份数据**
  - ** 在编译阶段分配内存**
  - ** 类内声明，类外初始化**
- 静态成员函数
  - ** 所有对象共享同一个函数**
  - ** 静态成员函数只能访问静态成员变量**

  接下来，我们用程序验证这个结论：

  1)首先是静态成员变量

```
class Person
{
public:
    static int m_A; //静态成员变量
    //静态成员变量特点：
    //1 在编译阶段分配内存
    //2 类内声明，类外初始化
    //3 所有对象共享同一份数据
private:
    static int m_B; //静态成员变量也是有访问权限的
};
int Person::m_A = 10;//类外初始化
int Person::m_B = 10;
void test01()
{
    //静态成员变量两种访问方式
    //1、通过对象
    Person p1;
    p1.m_A = 100;
    cout << "p1.m_A = " << p1.m_A << endl;
    Person p2;
    p2.m_A = 200;
    cout << "p1.m_A = " << p1.m_A << endl; //共享同一份数据
    cout << "p2.m_A = " << p2.m_A << endl;
    //2、通过类名
    cout << "m_A = " << Person::m_A << endl;
    //cout << "m_B = " << Person::m_B << endl; //私有权限访问不到
}
int main() {
    test01();
    system("pause");
    return 0;
}
```

输出：

```
p1.m_A = 100

p1.m_A = 200

p2.m_A = 200

m_A = 200
```

2)接下来是静态成员函数

```
class Person
{
public:
    //静态成员函数特点：
    //1 程序共享一个函数
    //2 静态成员函数只能访问静态成员变量
    static void func()
    {
        cout << "func调用" << endl;
        m_A = 100;
        //m_B = 100; //错误，不可以访问非静态成员变量
    }
    static int m_A; //静态成员变量
    int m_B; //
private:
    //静态成员函数也是有访问权限的
    static void func2()
    {
        cout << "func2调用" << endl;
    }
};
int Person::m_A = 10;
void test01()
{
    //静态成员变量两种访问方式
    //1、通过对象
    Person p1;
    p1.func();
    //2、通过类名
    Person::func();
    //Person::func2(); //私有权限访问不到
}
int main() {
    test01();
    system("pause");
    return 0;
}
```

输出：

```
func调用

func调用
```