

# 程序设计实训 HW1

姓名：王瀚森 学号：2024013325

## 作业一：类与对象

### 题目1：学生信息管理

#### 文件结构

- Member.h - 类声明
- Member.cpp - 类实现
- main.cpp - 测试程序

核心代码（部分代码为排版做了调整，并删除了注释，源代码麻烦助教还是查阅源程序）

#### Member.h

```
#pragma once
#include "Member.h"
#include <string>

class MemberList
{
public:
    MemberList();
    MemberList(Member* members, int size);

    std::string operator[](const std::string& name) const;

    ~MemberList();

private:
    Member* m_members;
    int m_size;
};
```

#### Member.cpp

```

#include "Member.h"
#include <iostream>
#include <string>

Member::Member() : m_name("?"), m_age(0)
{
}

Member::Member(std::string name, int age) : m_name(name), m_age(age)
{
}

Member::~Member()
{
}

const std::string Member::getName() const
{
    return m_name;
}

const int Member::getAge() const
{
    return m_age;
}

std::ostream& operator<<(std::ostream& os, const Member& m)
{
    os << "Name: " << m.getName() << ", Age: " << m.getAge();
    return os;
}

```

## main.cpp

```

#include "Member.h"
#include <iostream>
using namespace std;

int main()
{
    Member newCommers[5] = { Member("Zhang San", 22),
        Member("Li Si", 19),
        Member("Wang Wu", 18),
        Member("Zhao Liu", 24) };
    for (int i = 0; i < 5; i++)
    {
        cout << newCommers[i] << endl;
    }
    return 0;
}

```

## Visual Studio 编译运行步骤（之后的代码同理）

1. 创建新项目：文件→新建→项目→空项目
2. 添加源文件：右键“源文件”→添加→现有项→选择 Member.cpp, main.cpp
3. 添加头文件：右键“头文件”→添加→现有项→选择 Member.h
4. 编译运行：按 Ctrl+F5

## 运行结果

```
Name: Zhang San, Age: 22
Name: Li Si, Age: 19
Name: Wang Wu, Age: 18
Name: Zhao Liu, Age: 24
Name: ?, Age: 0
```

## 题目2：学生列表查询

### 文件结构

- Member.h - 类声明
- Member.cpp - 类实现
- MemberList.h - 列表类声明
- MemberList.cpp - 列表类实现
- main.cpp - 测试程序

### 核心代码

#### MemberList.h

```
#pragma once
#include "Member.h"
#include <string>

class MemberList
{
public:
    MemberList();
    MemberList(Member* members, int size);
    std::string operator[](const std::string& name) const;
    ~MemberList();
private:
    Member* m_members;
    int m_size;
};
```

## MemberList.cpp

```
#include "MemberList.h"
#include <iostream>

MemberList::MemberList(): m_members(nullptr), m_size(0)
{}

MemberList::MemberList(Member* members, int size) : m_members(
    ↪ members), m_size(size)
{}

std::string MemberList::operator[](const std::string& name) const
    ↪ {
    for (int i = 0; i < m_size; i++) {
        if (m_members[i].getName() == name) {
            return "The age of " + m_members[i].getName()
                + " is " + std::to_string(m_members[i].getAge());
        }
    }
    return "Student " + name + " is not found";
}

MemberList::~MemberList()
{}

```

## main.cpp

```
#include "Member.h"
#include "MemberList.h"
#include <iostream>
using namespace std;

int main()
{
    Member newCommers[5] = { Member("Zhang San", 22),
        Member("Li Si", 19),
        Member("Wang Wu", 18),
        Member("Zhao Liu", 24) };
    for (int i = 0; i < 5; i++)
    {
        cout << newCommers[i] << endl;
    }
    string name[5] = { "Zhang San", "Li Si", "Wang Wu", "Zhao Liu
        ↪ ", "Pin Yin" };
    MemberList list(newCommers, 5);
    for (int i = 0; i < 5; i++)
    {
        cout << list[name[i]] << endl;
    }
    return 0;
}

```

## 运行结果

```
Name: Zhang San, Age: 22
Name: Li Si, Age: 19
Name: Wang Wu, Age: 18
Name: Zhao Liu, Age: 24
Name: ?, Age: 0

The age of Zhang San is 22
The age of Li Si is 19
The age of Wang Wu is 18
The age of Zhao Liu is 24
Student Pin Yin is not found
```

## 作业二：继承与多态

### 题目：几何图形面积计算

#### 文件结构

- Shape.h - 基类和派生类声明
- Shape.cpp - 类实现
- main.cpp - 测试程序

#### 核心代码

**Shape.h**

```

#pragma once
class Shape
{
public:
    Shape();
    virtual ~Shape();
    virtual double getarea() const;
};
class Circle : public Shape
{
public:
    Circle(double radius);
    ~Circle();
    double getarea() const;
private:
    double m_radius;
};
class Rectangle : public Shape
{
public:
    Rectangle(double length, double width);
    ~Rectangle();
    double getarea() const;
private:
    double m_length;
    double m_width;
};
class Square : public Shape
{
public:
    Square(double sideLength);
    ~Square();
    double getarea() const;
private:
    double m_sideLength;
};

```

**Shape.cpp**

```

#include "Shape.h"

Shape::Shape(){}
Shape::~Shape(){}
double Shape::getarea() const
{
    return 0;
}
Circle::Circle(double radius): m_radius(radius)
{
}
Circle::~Circle()
{
}
double Circle::getarea() const
{
    return 3.1415926 * m_radius * m_radius;
}
Rectangle::Rectangle(double length, double width) : m_length(
    ↪ length), m_width(width)
{
}
Rectangle::~Rectangle()
{
}
double Rectangle::getarea() const
{
    return m_length * m_width;
}
Square::Square(double sideLength) : m_sideLength(sideLength)
{
}
Square::~Square()
{
}
double Square::getarea() const
{
    return m_sideLength * m_sideLength;
}

```

main.cpp

```

#include "Shape.h"
#include <iostream>
using namespace std;

int main()
{
    Shape* shapes[4];
    Circle circle(2.0);
    Rectangle rectangle(3.0, 4.0);
    Square square1(5.0);
    Square square2(4.0);
    shapes[0] = &circle;
    shapes[1] = &rectangle;
    shapes[2] = &square1;
    shapes[3] = &square2;
    for (int k = 0; k < 4; k++) {
        cout << "Area is " << shapes[k]->getarea() << endl;
    }
    return 0;
}

```

运行结果

```

Area is 12.5664
Area is 12
Area is 25
Area is 16

```

## 作业三：模板与特化

题目：通用最大值函数

文件结构

- TemplateMax.h - 模板函数声明和实现
- main.cpp - 测试程序

核心代码

TemplateMax.h



```

#pragma once
#include <string>
#include <cstring>

template <typename T>
T Max(T x, T y) {
    if (x > y) {
        return x;
    }
    return y;
}

const char* Max(const char* x, const char* y) {
    if (strcmp(x, y) > 0)
        return x;
    return y;
}

```

#### main.cpp

```

#include "TemplateMax.h"
#include <string>
#include <iostream>
using namespace std;

int main()
{
    int i = 1;
    int j = 2;
    cout << "Max(i, j): " << Max(i, j) << endl;
    double f1 = 11.1;
    double f2 = 22.2;
    cout << "Max(f1, f2): " << Max(f1, f2) << endl;
    string s1 = "AAAAA";
    string s2 = "BBBBB";
    cout << "Max(s1, s2): " << Max(s1, s2) << endl;
    return 0;
}

```

#### 运行结果

```

Max(i, j): 2
Max(f1, f2): 22.2
Max(s1, s2): BBBBB

```