

编号: 3-1



山东师范大学
SHANDONG NORMAL UNIVERSITY

信息科学与工程学院实验报告

《面向对象程序设计》

Object-Oriented Programming

姓名:	张泽浩
学号:	202111000212
班级:	计工本 2102
导师:	张庆科
时间:	2021 年 10 月 29 日



《面向对象程序设计》实验报告

基本要求：实验报告包含实验目的、实验内容、实验过程（详细操作流程）、实验结果（程序运行结果高清截图）、实验分析总结五个部分。报告中若涉及代码程序，请在附录部分提供完整程序源码及源码托管地址(基于 Highlight 软件导入源码)。报告撰写完毕后请提交 PDF 格式版本报告到课程云班课系统。

一、实验目的

1. 理解类和对象的基本概念
2. 掌握类的定义和实现方法
3. 掌握类内函数两种实现方法
4. 能够独立在 VS 环境下设计类

二、实验内容

（一）任务一：设计并实现一个学生类

建立 VS C++项目，设计并实现一个学生类 **Student**，类内学生信息属性自拟，该类对外提供的访问接口包括：初始化学生信息、获取学生信息、修改学生信息、输出学生信息。

（二）任务二：设计并实现一个三角形类

建立 VS C++项目，设计并实现一个三角形类 **Triangle**，该类对外提供的接口包括：初始化三角形的各条边（要求使用三种构造函数实现）、修改三角形对象边长、计算三角形对象的周长、计算三角形对象的面积、判断三角形对象的类型（直角，锐角还是钝角）。

三、实验过程

（一）任务一：设计并实现一个学生类

1. 学生类包含的属性

(1) 访问属性：私有成员 `private`

(2) 属性：姓名、性别、学号、年龄、C 语言/高数/英语成绩（如图 1）

```
private:
    string name;           // 姓名
    char sex;              // 性别(f:男 m:女)
    string ID;             // 学号
    int age;               // 年龄
    double CScore;         // C语言成绩
    double MathScore;      // 高数成绩
    double EnglishScore;   // 英语成绩
```

图 1 学生信息类中的属性



2. 学生类包含的行为

- (1) 访问属性：公有成员函数 `public`
- (2) 成员函数：初始化学生信息、获取学生信息、修改学生信息、输出学生信息。

(二) 任务二：设计并实现一个三角形类

1. 三角形类包含的属性

- (1) 访问属性：私有成员 `private`
- (2) 属性：三角形三边边长（如图 2）

```
private:
    double a, b, c;
```

图 2 三角形类中的属性

2. 三角形类包含的行为

- (1) 访问属性：私有属性 `private`、保护属性 `protected`、公有属性 `public`
- (2) 成员函数：初始化三角形的各条边（三类构造函数如图 3）、修改三角形对象边长、计算三角形对象的周长、计算三角形对象的面积、判断三角形对象的类型

```
private:
    void set_a(double _a);
protected:
    void set_b(double _b);
public:
    void set_c(double _c);
```

图 3 初始化三角形各边的三类构造函数

四、实验结果

(一) 三类界面

1. 主界面



图 4 主界面运行截图



2. 学生信息类主界面



图 5 学生信息类主界面运行结果

3. 三角形类主界面



图 6 三角形类主界面运行结果

(二) 学生信息类

1. 测试数据

表 1 学生信息测试案例

姓名	性别	年龄	学号	C 语言成绩	高数成绩	英语成绩
Zehao	男	19	202111000212	99	100	98

2. 运行结果



图 7 输出学生信息运行结果



(三) 三角形类

1. 测试案例

表 2 三角形类测试案例

边长 1	边长 2	边长 3	周长	面积	三角形类型
3	4	5	12	6	直角三角形

表 3 修改后的三角形测试案例

边长 1	边长 2	边长 3	周长	面积	三角形类型
6	8	12	26	21.3307	钝角三角形

2. 运行结果

```
D:\C & C++\Object-oriented-Programming\OOP_Experiment\Experiment-3_1\E...
--
--          三角形类          --
--
--      1. 修改三角形边长      --
--      2. 计算三角形的周长    --
--      3. 计算三角形的面积    --
--      4. 判断三角形的类型    --
--      0. 退出三角形类      --
--
请选择您要测试的选项: 2
三角形周长为: 12
请按任意键继续. . .
```

图 8 计算三角形周长运行结果

```
D:\C & C++\Object-oriented-Programming\OOP_Experiment\Experiment-3_1\E...
--
--          三角形类          --
--
--      1. 修改三角形边长      --
--      2. 计算三角形的周长    --
--      3. 计算三角形的面积    --
--      4. 判断三角形的类型    --
--      0. 退出三角形类      --
--
请选择您要测试的选项: 3
三角形面积为: 6
请按任意键继续. . .
```

图 9 计算三角形面积运行结果

```
D:\C & C++\Object-oriented-Programming\OOP_Experiment\Experiment-3_1\E...
--
--          三角形类          --
--
--      1. 修改三角形边长      --
--      2. 计算三角形的周长    --
--      3. 计算三角形的面积    --
--      4. 判断三角形的类型    --
--      0. 退出三角形类      --
--
请选择您要测试的选项: 4
三角形类型为: 直角三角形
请按任意键继续. . .
```

图 10 判断三角形类型运行结果



```
D:\C & C++\Object-oriented-Programming\OOP_Experiment\Experiment-3_1\E...
--
--      三角形类      --
--
--      1. 修改三角形边长      --
--      2. 计算三角形的周长      --
--      3. 计算三角形的面积      --
--      4. 判断三角形的类型      --
--      0. 退出三角形类      --
--
请选择您要测试的选项: 1
请输入新的三边边长: 6 8 12
修改成功!
请按任意键继续. . .
```

图 11 修改三角形边长信息运行结果

五、实验总结

通过本实验对学生类与三角形类的实现，掌握了类的定义和实现方法，理解类和对象的基本概念，并且能够独立设计类。在实验中，还通过 `while` 循环和 `switch` 语句实现了功能的选择，通过 `system("pause")` 和 `system("cls")` 实现了界面的转换。



附录：实验源代码（基于 Highlight 软件粘贴带有行号的源码）

代码托管地址：

[Object-oriented-Programming/OOP_Experiment/Experiment-3_1/Experiment-3_1 at master · keepIHDR/Object-oriented-Programming \(github.com\)](https://github.com/keepIHDR/Object-oriented-Programming/blob/master/Experiment-3_1/Experiment-3_1_at_master/keepIHDR/Object-oriented-Programming)

main.cpp

```
01 #include <iostream>
02 #include "triangle.h"
03 #include "student.h"
04
05 using namespace std;
06
07 void showAll() {
08     cout << "-----" << endl;
09     cout << "--      实验3_1: 类的设计与实现      --" <
10     < endl;
11     cout << "--      1. 学生信息类      --" <<
12     endl;
13     cout << "--      2. 三角形类      --" <<
14     endl;
15     cout << "--      0. 退出程序      --" <<
16     endl;
17     cout << "-----" << endl;
18 }
19
20 int main() {
21     int select = 0;
22     while (1) {
23         system("cls");
24         showAll();
25         cout << "请选择您要测试的类: ";
26         cin >> select;
27         switch (select) {
28             case 1:
29                 Class_Stu();
30                 break;
31             case 2:
32                 Class_Triangle();
33                 break;
34             case 0:
35                 system("pause");
36                 return 0;
37             default:
38                 cout << "输入选项错误!" << endl;
39                 system("pause");
```



```
40         break;
41     }
42 }
43 }

学生类
student.h
01 #pragma once
02 #include <string>
03 using namespace std;
04
05 class Student {
06     private:
07         string name;           // 姓名
08         char sex;              // 性别(f:男 m:女)
09         string ID;             // 学号
10         int age;               // 年龄
11         double CScore;         // C语言成绩
12         double MathScore;      // 高数成绩
13         double EnglishScore;   // 英语成绩
14     public:
15         void Init(string Nname, char Nsex, string NID, int Nage,
16                 double NCScore, double NMathScore, double
17                 NEnglishScore);
18         // 初始化学生信息
19         string getName();       // 获取姓名
20         string getSex();        // 获取性别
21         string getID();         // 获取学号
22         int getAge();           // 获取年龄
23         double getCScore();     // 获取C语言成绩
24         double getMathScore();  // 获取高数成绩
25         double getEnglishScore(); // 获取英语成绩
26         void PrintInformation(); // 输出学生信息
27         void Modify_name(string Nname); // 修改学生姓名
28         void Modify_sex(char Nsex); // 修改学生性别
29         void Modify_ID(string NID); // 修改学生学号
30         void Modify_age(int Nage); // 修改学生年龄
31         void Modify_CScore(double NCScore); // 修改学生C语言成绩
32         void Modify_MScore(double NMathScore); // 修改学生高数成绩
33         void Modify_EScore(double NEnglishScore); // 修改学生英语成绩
41 };
42
43 void showStu();
44 void getSTU();
```




```
45 void ModifySTU();
46 void Class_Stu();

student.cpp
01 #include "student.h"
02 #include <iostream>
03
04 using namespace std;
05
06 void Student::Init(string Nname, char Nsex, string NID, int
07 Nage, double NCscore, double NMathScore, double
08 NEnglishScore) {
09     name = Nname;
10     sex = Nsex;
11     ID = NID;
12     age = Nage;
13     CScore = NCscore;
14     MathScore = NMathScore;
15     EnglishScore = NEnglishScore;
16 }
17
18 string Student::getName() {
19     return name;
20 }
21
22 string Student::getSex() {
23     if (sex == 'f') {
24         return "男";
25     } else {
26         return "女";
27     }
28 }
29
30 string Student::getID() {
31     return ID;
32 }
33
34 int Student::getAge() {
35     return age;
36 }
37
38 double Student::getCScore() {
39     return CScore;
40 }
```



```
41
42 double Student::getMathScore() {
43     return MathScore;
44 }
45
46 double Student::getEnglishScore() {
47     return EnglishScore;
48 }
49
50 void Student::PrintInformation() {
51     cout << ">> " << getName() << "同学的信息如下: " <
52     < endl;
53     cout << "    性别: \t" << getSex() << endl;
54     cout << "    年龄: \t" << getAge() << endl;
55     cout << "    学号: \t" << getID() << endl;
56     cout << "    C语言成绩: \t" << getCScore() << endl;
57     cout << "    高数成绩: \t" << getMathScore() << endl;
58     cout << "    英语成绩: \t" << getEnglishScore() <<
59     endl;
60 }
61
62 void Student::Modify_name(string Nname) {
63     name = Nname;
64     cout << "修改成功!" << endl;
65 }
66
67 void Student::Modify_sex(char Nsex) {
68     sex = Nsex;
69     cout << "修改成功!" << endl;
70 }
71
72 void Student::Modify_ID(string NID) {
73     ID = NID;
74     cout << "修改成功!" << endl;
75 }
76
77 void Student::Modify_age(int Nage) {
78     age = Nage;
79     cout << "修改成功!" << endl;
80 }
81
82 void Student::Modify_CScore(double NCScore) {
83     CScore = NCScore;
84     cout << "修改成功!" << endl;
```



```
85 }
86
87 void Student::Modify_MScore(double NMathScore) {
88     MathScore = NMathScore;
89     cout << "修改成功!" << endl;
90 }
91
92 void Student::Modify_EScore(double NEnglishScore) {
93     EnglishScore = NEnglishScore;
94     cout << "修改成功!" << endl;
95 }
96
97 Student s;
98
99 void showStu() {
100     cout << "-----" << endl;
101     cout << "--      学生信息类      --" <<
102     endl;
103     cout << "-----" << endl;
104     cout << "--      1.获取学生信息      --" <<
105     endl;
106     cout << "--      2.修改学生信息      --" <<
107     endl;
108     cout << "--      3.输出学生信息      --" <<
109     endl;
110     cout << "--      0.退出学生类      --" <<
111     endl;
112     cout << "-----" << endl;
113     cout << "请选择您要测试的选项：";
114 }
115
116
117 void getSTU() {
118     cout << "-----" << endl;
119     cout << "--      获取学生信息      --" <<
120     endl;
121     cout << "-----" << endl;
122     cout << "--      1.查看学生姓名      --" <<
123     endl;
124     cout << "--      2.查看学生性别      --" <<
125     endl;
126     cout << "--      3.查看学生年龄      --" <<
127     endl;
128     cout << "--      4.查看学生学号      --" <<
```



```
129     endl;
130     cout << "--          5.查看学生C语言成绩          --" <<
131     endl;
132     cout << "--          6.查看学生高数成绩          --" <<
133     endl;
134     cout << "--          7.查看学生英语成绩          --" <<
135     endl;
136     cout << "-----" << endl;
137     cout << "请选择您要查看的学生信息：";
138 }
139
140 void ModifySTU() {
141     cout << "-----" << endl;
142     cout << "--          修改学生信息          --" <<
143     endl;
144     cout << "-----" << endl;
145     cout << "--          1.修改学生姓名          --" <<
146     endl;
147     cout << "--          2.修改学生性别          --" <<
148     endl;
149     cout << "--          3.修改学生年龄          --" <<
150     endl;
151     cout << "--          4.修改学生学号          --" <<
152     endl;
153     cout << "--          5.修改学生C语言成绩          --" <<
154     endl;
155     cout << "--          6.修改学生高数成绩          --" <<
156     endl;
157     cout << "--          7.修改学生英语成绩          --" <<
158     endl;
159     cout << "-----" << endl;
160     cout << "请选择您要修改的学生信息：";
161 }
162
163 void Class_Stu() {
164     int temp1 = 0;
165     int temp2 = 0;
166     int select = 0;
167     string name;
168     char sex = 0;
169     string ID;
170     int age = 0;
171     double score = 0.0;
172     s.Init("Zehao", 'f', "202111000212", 19, 99, 100, 98);
```



```
173     system("cls");
174     cout << "初始化学生信息成功!" << endl;
175     system("pause");
176     while (1) {
177         system("cls");
178         showStu();
179         cin >> select;
180         switch (select) {
181             case 1:
182                 system("cls");
183                 getSTU();
184                 cin >> temp1;
185                 switch (temp1) {
186                     case 1:
187                         cout << "学生的姓名为: " << s.getName()
188                             << endl;
189                         break;
190                     case 2:
191                         cout << "学生的性别为: " << s.getSex()
192                             << endl;
193                         break;
194                     case 3:
195                         cout << "学生的年龄为: " << s.getAge()
196                             << endl;
197                         break;
198                     case 4:
199                         cout << "学生的学号为: " << s.getID() <
200                             < endl;
201                         break;
202                     case 5:
203                         cout << "学生的C语言成绩为: " << s.
204                             getCScore() << endl;
205                         break;
206                     case 6:
207                         cout << "学生的高数成绩为: " << s.
208                             getMathScore() << endl;
209                         break;
210                     case 7:
211                         cout << "学生的英语成绩为: " << s.
212                             getEnglishScore() << endl;
213                         break;
214                     default:
215                         cout << "输入的选项错误!" << endl;
216                         break;
```



```
217     }
218     break;
219 case 2:
220     system("cls");
221     ModifySTU();
222     cin >> temp2;
223     switch (temp2) {
224     case 1:
225         cout << "学生的姓名更改为: ";
226         cin >> name;
227         s.Modify_name(name);
228         break;
229     case 2:
230         cout << "学生的性别更改为(f:男 m:
231         女): ";
232         cin >> sex;
233         s.Modify_sex(sex);
234         break;
235     case 3:
236         cout << "学生的年龄更改为: ";
237         cin >> age;
238         s.Modify_age(age);
239         break;
240     case 4:
241         cout << "学生的学号更改为: ";
242         cin >> ID;
243         s.Modify_ID(ID);
244         break;
245     case 5:
246         cout << "学生的C语言成绩更改为: ";
247         cin >> score;
248         s.Modify_CScore(score);
249         break;
250     case 6:
251         cout << "学生的高数成绩更改为: ";
252         cin >> score;
253         s.Modify_MScore(score);
254         break;
255     case 7:
256         cout << "学生的英语成绩更改为: ";
257         cin >> score;
258         s.Modify_EScore(score);
259         break;
260     default:
```



```
261         cout << "输入的选项错误!" << endl;
262         break;
263     }
264     break;
265     case 3:
266         s.PrintInformation();
267         break;
268     case 0:
269         return;
270     default:
271         cout << "输入选项错误!" << endl;
272         break;
273     }
274     system("pause");
275 }
276 }
```

三角形类

triangle.h

```
01 #pragma once
02 #include <string>
03 #include <iostream>
04
05 using namespace std;
06
07 class triangle {
08     // 属性：数据
09     private:
10         double a, b, c;
11     // 行为：函数
12     private:
13         void set_a(double _a);
14     protected:
15         void set_b(double _b);
16     public:
17         void set_c(double _c);
18         void PrintInformation(); // 输出三角形相关信息
19         void Init(double newa, double newb, double newc); // 初始化
20         void Modify_abc(double Na, double Nb, double Nc); // 修改边长
21         bool isTriangle(); // 判断是否能够构成三角形
22         double length(); // 计算周长
23         double area(); // 计算面积
24         string typeTriangle(); // 判断三角形类型
25     private:
```



```
28 void SortTriangle();// 三边排序
29 };
30
31 void showTriangle();
32 void Class_Triangle();
33
34
35 triangle.cpp
36
37 01 #include "triangle.h"
38 02 #include <cmath>
39 03 #include <iostream>
40 04
41 05 using namespace std;
42 06
43 07 void triangle::set_a(double _a) {
44 08     a = _a;
45 09 }
46 10
47 11 void triangle::set_b(double _b) {
48 12     b = _b;
49 13 }
50 14
51 15 void triangle::set_c(double _c) {
52 16     c = _c;
53 17 }
54 18
55 19 void triangle::PrintInformation() {
56 20     cout << ">> 该三角形的信息如下: " << endl;
57 21     cout << "    三边边长为: \t\t" << a << ", " << b << ",
58 22     " << c << endl;
59 23     cout << "    三角形的周长为: \t" << length() <<
60 24     endl;
61 25     cout << "    三角形的面积为: \t" << area() << endl;
62 26     cout << "    三角形的类型为: \t" << typeTriangle()
63 27     << endl;
64 28 }
65 29
66 30 void triangle::Init(double newa, double newb, double newc) {
67 31     set_a(newa);
68 32     set_b(newb);
69 33     set_c(newc);
70 34     SortTriangle();
71 35 }
72 36
73 37 void triangle::Modify_abc(double Na, double Nb, double Nc) {
```




```
38     if (Na + Nb > Nc && Na + Nc > Nb && Nb + Nc > Na) {
39         a = Na;
40         b = Nb;
41         c = Nc;
42         SortTriangle();
43         cout << "修改成功!" << endl;
44     } else {
45         cout << "新的三边不能构成三角形!" << endl;
46         return;
47     }
48 }
49
50 bool triangle::isTriangle() {
51     if (a + b > c && a + c > b && b + c > a) {
52         return true;
53     } else {
54         return false;
55     }
56 }
57
58 double triangle::length() {
59     return a + b + c;
60 }
61
62 double triangle::area() {
63     double p = 0.5 * (a + b + c);
64     return sqrt(p * (p - a) * (p - b) * (p - c));
65 }
66
67 string triangle::typeTriangle() {
68     if (a * a + b * b > c * c) {
69         return "锐角三角形";
70     } else if (a * a + b * b == c * c) {
71         return "直角三角形";
72     } else {
73         return "钝角三角形";
74     }
75 }
76
77 void triangle::SortTriangle() {
78     if (a > b) {
79         int temp = a;
80         a = b;
81         b = temp;
```



```
82     }
83     if (b > c) {
84         int temp = b;
85         b = c;
86         c = temp;
87     }
88     if (a > b) {
89         int temp = a;
90         a = b;
91         b = temp;
92     }
93 }
94
95 triangle A;
96
97 void showTriangle() {
98     cout << "-----" << endl;
99     cout << "--          三角形类          --" <<
100     endl;
101     cout << "-----" << endl;
102     cout << "--          1.修改三角形边长          --" <<
103     endl;
104     cout << "--          2.计算三角形的周长          --" <<
105     endl;
106     cout << "--          3.计算三角形的面积          --" <<
107     endl;
108     cout << "--          4.判断三角形的类型          --" <<
109     endl;
110     cout << "--          0.退出三角形类          --" <<
111     endl;
112     cout << "-----" << endl;
113     cout << "请选择您要测试的选项：";
114 }
115
116 void Class_Triangle() {
117     int select = 0;
118     int a, b, c;
119     A.Init(3, 4, 5);
120     system("cls");
121     cout << "初始化三角形成功!" << endl;
122     system("pause");
123     while (1) {
124         system("cls");
125         showTriangle();
```



```
126     cin >> select;
127     switch (select) {
128     case 1:
129         cout << "请输入新的三边边长: ";
130         cin >> a >> b >> c;
131         A.Modify_abc(a, b, c);
132         break;
133     case 2:
134         cout << "三角形周长为: " << A.length() <<
135         endl;
136         break;
137     case 3:
138         cout << "三角形面积为: " << A.area() <<
139         endl;
140         break;
141     case 4:
142         cout << "三角形类型为: " << A.typeTriangle(
143         ) << endl;
144         break;
145     case 0:
146         return;
147     default:
148         cout << "输入选项错误!" << endl;
149         break;
150     }
151     system("pause");
152 }
153 }
```