面向对象程序设计

笔记整理与实验

**学院： 信息与电气工程学院**

**学号： 20222102972**

**班级： 软工2202**

**姓名： 张瑞哲**

**教师： 苏兆锋**

# 目录

目录

[目录 2](#_Toc135861720)

[一．复杂模块的构造与解体 3](#_Toc135861721)

[1.复杂模块（类）的组成构造与解体意义和语法关联理解 3](#_Toc135861722)

[复杂模块（类）的组成 4](#_Toc135861723)

[2.代码实现与运行截图 5](#_Toc135861724)

[2.1 5](#_Toc135861725)

[2.2 7](#_Toc135861726)

[2.3 10](#_Toc135861727)

[2.4 14](#_Toc135861728)

[2.5 16](#_Toc135861729)

[2.6 19](#_Toc135861730)

[3.领悟与反思 22](#_Toc135861731)

[二．模块拓展：继承与派生 22](#_Toc135861732)

[1. 模块拓展（继承与派生）的理解和语法关联理解 23](#_Toc135861733)

[继承 23](#_Toc135861734)

[友元函数、友元类 25](#_Toc135861735)

[数据模块存储 26](#_Toc135861736)

[模块的内存分配 26](#_Toc135861737)

[浅拷贝、深拷贝 27](#_Toc135861738)

[数据的保护 27](#_Toc135861739)

[2.代码实现与运行截图 28](#_Toc135861740)

[2.1 28](#_Toc135861741)

[2.2 31](#_Toc135861742)

[2.3 33](#_Toc135861743)

[2.4 37](#_Toc135861744)

[2.5 41](#_Toc135861745)

[2.6 43](#_Toc135861746)

[2.7 46](#_Toc135861747)

[2.8 49](#_Toc135861748)

[2.9 54](#_Toc135861749)

[3.领悟与反思 55](#_Toc135861750)

[三．模块类型认同：多态 55](#_Toc135861751)

[四．其他部分 55](#_Toc135861752)

# 一．复杂模块的构造与解体

## 1.复杂模块（类）的组成构造与解体意义和语法关联理解

功能模块：能够相对独立出来作一个单元  
一个与外界相对独立的单元可以理解为模块，这个模块对外界具备一定的功能，我们可以称为功能模块。任何一个大模块（系统）都是由子模块的集合按照一个结构构成，向外展现出一定的功能。

解决问题的关键是分解问题，核心是用功能化、模块化思想去分解问题

如何功能化模块化：自顶向下，逐步细化，将小功能作为一个独立功能单元

开始时要简单问题复杂化

将算法转换为代码  
void function() {  
// 算法思想：把大象关进冰箱分为三步  
 // 1. 打开冰箱门  
 func1(open door)  
 ….  
}

自顶向下，逐步累积，小模块积累为大模块  
我们对事物全局没有全面的认识时，就需要从小到向大，由点及面进行扩展，以至最终认识事物。

模块的结构与构建（模块构建要提供构造说明书）  
模块名 {  
 子元素个体集合  
 子元素关系  
}

封装和接口（模块元素对内外区别）  
一个模块通用定义  
模块名 {  
 private:  
 私有部分/隐藏部分

public:  
 接口部分  
}

模块：基础元素+新增元素集  
要表现出的基础特性 public

构成 {  
 元素：  
 a b c d  
 构造函数（aa bb cc dd）  
 功能：  
 e f  
}

结构 {  
 对内：  
 a b c d  
 对外：  
 e f  
}

手表 {  
内部元素：  
 获得时间模块  
对外模块：  
 显示时间  
 显示时间：时针分针秒针 表盘 解读方式  
 子元素关系  
}

### 复杂模块（类）的组成

#### 类名

类名是一个标识符，用于标识该类的存在。类名通常以大写字母开头。

2.

## 2.代码实现与运行截图

### 2.1

#### 题目：设计一个学生信息类

##### 任务描述

本关任务：声明并定义一个学生信息类。

##### 编程要求

在右侧编辑器中的Begin-End之间补充代码，声明并定义一个学生信息类 StInfo，该类有四个成员变量和两个函数，其访问性都为公有，具体要求如下：

* 学号：int SID
* 姓名：char \*Name
* 班级：char \*Class
* 手机号：char \*Phone
* 给成员变量赋值的函数： void SetInfo(int sid,char \*name,char\* cla,char\* phone)
* 打印学生信息的函数：void PrintInfo()，格式请参考测试说明。

##### 测试说明

平台会对你编写的代码进行测试，比对你输出的数值与实际正确数值，只有所有数据全部计算正确才能通过测试：

测试输入：1 小郭 计科1班 12312340000 预期输出：

1. 学号：1
2. 姓名：小郭
3. 班级：计科1班
4. 手机号：12312340000

测试输入：2 小王 计科2班 12312340001 预期输出：

1. 学号：2
2. 姓名：小王
3. 班级：计科2班
4. 手机号：12312340001

#### 代码与运行

#include <iostream>

using namespace std;

class StInfo

{

/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

//在此处声明StInfo类

private:

int SID;

string name, cls, phone;

public:

void SetInfo(int sid, char \*name, char \*cls, char \*phone);

void PrintInfo();

/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

};

/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

//在此处定义StInfo类

void StInfo::SetInfo(int sid, char \*name, char \*cls, char \*phone) {

this->SID = sid;

this->name = name;

this->cls = cls;

this->phone = phone;

}

void StInfo::PrintInfo() {

std::cout << "学号：" << this->SID << '\n';

std::cout << "姓名：" << this->name << '\n';

std::cout << "班级：" << this->cls << '\n';

std::cout << "手机号：" << this->phone << '\n';

}

/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

### 2.2

#### 题目：设计一个长方形类

##### 任务描述

本关任务：设计一个长方形类，通过访问它的成员变量来计算面积。

##### 编程要求

在右侧编辑器中的Begin-End之间补充代码，设计 Rectangle 类和实现两个普通函数，类中有两个成员变量和两个函数，其中成员变量的访问性为私有，函数的访问性为公有，具体要求如下：

* 高度：int height
* 宽度：int width
* 设置长方形的高和宽函数：void Set(int h, int w)
* 获取长方形的面积函数：int GetArea()
* Rectangle GetRect(int h,int w)普通函数，h 、w 分别代表长方形的高宽，函数用来创建一个 Rectangle 对象并返回。
* int GetRectArea(Rectangle rect)普通函数，函数用来获取 rect 对象的面积。

##### 测试说明

平台会对你编写的代码进行测试，比对你输出的数值与实际正确数值，只有所有数据全部计算正确才能通过测试：

测试输入：10 15 预期输出：长方形的面积为：150

测试输入：100 100 预期输出：长方形的面积为：10000

#### 代码与运行

/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

class Rectangle

{

//在此处实现Rectangle类

public:

int height, width;

Rectangle(int h, int w) {

this->height = h;

this->width = w;

}

};

/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

Rectangle GetRect(int h,int w)

{

/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

//返回一个 h\*w 的 Rectangle 对象

Rectangle t(h, w);

return t;

/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

}

int GetRectArea(Rectangle rect)

{

/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

//返回 rect 对象的面积

return rect.height \* rect.width;

/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

}

### 2.3

#### 题目：设计汽车类

##### 任务描述

本关任务：设计一个汽车类，并实现它的基本功能。

##### 编程要求

在右侧编辑器中的Begin-End之间补充代码，设计汽车类实现汽车的基本功能，根据输入的命令来输出汽车的整体状态。具体要求如下：

1. 成员变量：车门，车灯，速度（类型以及变量名均自拟）。
2. 成员函数：打开车门、关闭车门、打开车灯、关闭车灯、加速、减速。（学员可以根据自己情况增加或者减少函数）

其中打开车门、关闭车门、打开车灯、关闭车灯四个函数，用于改变对象内部对应的成员变量的值。加速，减速两个函数，每次调用就将内部的速度变量分别对应增加10或者减少10。车门，车灯两个函数的初始状态都为 OFF，速度的初始值为0。

根据输入的命令（命令由1-6表示，分别对应调用打开车门、关闭车门、打开车灯、关闭车灯、加速、减速函数）输出汽车的最终状态，若输入命令135则输出状态如下：

1. 车门 ON
2. 车灯 ON
3. 速度 10

上述命令为135，则分表调用了打开车门、打开车灯、加速这三个函数，因此最后输出的汽车状态为车门打开，车灯打开和速度为10。

##### 测试说明

平台会对你编写的代码进行测试，比对你输出的数值与实际正确数值，只有所有数据全部计算正确才能通过测试：

注意：命令长度不超过 20，最短长度 1。

测试输入：135 预期输出：

1. 车门 ON
2. 车灯 ON
3. 速度 10

测试输入：135562 预期输出：

1. 车门 OFF
2. 车灯 ON
3. 速度 10

#### 代码与运行

*#include <iostream>*

*#include <cstring>*

*using namespace std;*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*//在此处实现一个汽车类*

*class Car*

*{*

*public:*

*int door = 0, light = 0, speed = 0;*

*void open\_door() {*

*door = 1;*

*}*

*void close\_door() {*

*door = 0;*

*}*

*void open\_light() {*

*light = 1;*

*}*

*void close\_light() {*

*light = 0;*

*}*

*void speed\_up() {*

*speed += 10;*

*}*

*void speed\_down() {*

*speed -= 10;*

*}*

*};*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*int main()*

*{*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*//在此处解析执行输出汽车的整体状态*

*char cmds[25];*

*cin>>cmds;*

*int n = strlen(cmds);*

*Car car;*

*for (int i = 0; i < n; i++) {*

*if (cmds[i] == '1') {*

*car.open\_door();*

*}*

*if (cmds[i] == '2') {*

*car.close\_door();*

*}*

*if (cmds[i] == '3') {*

*car.open\_light();*

*}*

*if (cmds[i] == '4') {*

*car.close\_light();*

*}*

*if (cmds[i] == '5') {*

*car.speed\_up();*

*}*

*if (cmds[i] == '6') {*

*car.speed\_down();*

*}*

*}*

*cout << "车门 " << (car.door ? "ON" : "OFF") << endl;*

*cout << "车灯 " << (car.light ? "ON" : "OFF") << endl;*

*cout << "速度 " << car.speed << endl;*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*}*

### 2.4

#### 题目：构造函数 —— 学生信息类

##### 任务描述

本关任务：设计一个带有构造函数和析构函数的学生类。

##### 编程要求

在右侧编辑器中的Begin-End之间补充代码，编写一个学生类 Student，类中含有两个成员变量、两个构造函数和一个析构函数（访问性都为公有的），平台会调用你编写的函数来创建学生对象，具体要求如下：

* 学号：int SID
* 姓名：string Name
* 无参的构造函数：Student()，在函数中将学号初始化为0，姓名初始化为王小明。
* 带两个参数的构造函数：Student(int sid,string name);，在函数中分别用这两个参数设置内部的两个成员。
* 析构函数：~Student()，在函数中输出学号 姓名 退出程序的消息。

##### 测试说明

平台会对你编写的代码进行测试，比对你输出的数值与实际正确数值，只有所有数据全部计算正确才能通过测试：

测试输入：1 厉宏富 2 冷欣荣 3 鲍俊民

预期输出：

1. 0 王小明 退出程序
2. 3 鲍俊民 退出程序
3. 2 冷欣荣 退出程序
4. 1 厉宏富 退出程序

#### 代码与运行

*#include<iostream>*

*#include<string>*

*using namespace std;*

*class Student*

*{*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*//在此处声明所需的成员*

*public:*

*int SID;*

*string Name;*

*Student(int, string);*

*~Student();*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*};*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*//在此处定义成员函数*

*Student::Student(int SID = 0, string Name = "王小明") : SID(SID), Name(Name) {*

*return;*

*}*

*Student::~Student() {*

*std::cout << SID << ' ' << Name << ' ' << "退出程序" << endl;*

*}*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

### 2.5

#### 题目：对象数组 —— 学生信息表

##### 任务描述

本关任务：编写一个能管理多条学生信息的程序。

##### 编程要求

在右侧编辑器中的Begin-End之间补充代码，设计 Student 类并实现用于管理学生信息表（学生表的长度不超过5）的3个函数，成员变量和函数的访问性都为公有的，具体类结构和函数要求如下：

* 学号，int类型
* 姓名，string类型
* 分数，float类型
* 带参构造函数：Student(int sid,string name,float sco)，分别用这三个参数设置内部的三个成员。
* void Add(int sid,string name,float sco)，函数用于向学生表的末尾添加一条学生记录。
* void PrintAll()，输出学生表中所有的记录，格式为：学号 姓名 成绩。
* void Average()，计算学生表中学生的平均成绩并输出，格式为：平均成绩 计算结果。

**提示**：学生表可以用全局对象数组来完成，定义全局对象数组和定义全局变量一样，即定义在最外层作用域。

##### 测试说明

平台会对你编写的代码进行测试，比对你输出的数值与实际正确数值，只有所有数据全部计算正确才能通过测试：

测试输入：0 厉宏富 96 1 冷欣荣 85 2 鲍俊民 76

预期输出：

1. 0 厉宏富 96
2. 1 冷欣荣 85
3. 2 鲍俊民 76
4. 平均成绩 85.6667

*代码与运行*

*#include <iostream>*

*#include <string>*

*using namespace std;*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*class Student {*

*// 在此处声明所需的成员*

*public:*

*int SID;*

*string name;*

*double score;*

*Student() {*

*}*

*Student(int sid, string name, float sco) {*

*this->SID = sid;*

*this->name = name;*

*this->score = sco;*

*}*

*};*

*Student a[100];*

*int n = 0;*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*void Add(int sid, string name, float sco) {*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*a[++n] = Student(sid, name, sco);*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*}*

*void PrintAll() {*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*// 打印出学生表中所有记录*

*for (int i = 1; i <= n; i++) {*

*cout << a[i].SID << ' ' << a[i].name << ' ' << a[i].score << endl;*

*}*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*}*

*void Average() {*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*// 计算并打印出学生表中的平均成绩*

*float avg = 0;*

*for (int i = 1; i <= n; i++) {*

*avg += a[i].score;*

*}*

*std::cout << "平均成绩 " << avg / n << endl;*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

*}*

### 2.6

#### 题目：静态成员 —— 模拟共享书店

##### 任务描述

本关任务：假设有一个这样的共享书店，当客户进入书店时需要托管一定数量的书籍，而这些书籍将被书店内所有用户共享，当客户离开书店时，他便会把之前进入书店托管的书一起带走。现请你编写代码设计这个共享书店。

##### 编程要求

在右侧编辑器中的Begin-End之间补充代码，设计一个 User 类（客户类），现有一个共享书店，该书店客户在进入书店时需要托管一定量的书籍，而这些书籍将由书店内所有用户共享，当客户离开书店时，他还是要将他进入书店时托管的书带走。设计时访问性可自主选择，具体要求如下：

* 姓名：string Name
* 托管的书籍量：int Books
* 带参构造函数：User(string name,int books)，使用这两个参数初始化内部的两个成员，同时按照姓名 数量 进入的格式打印一条消息。
* 析构函数：~User()，按照姓名 数量 离开的格式打印一条消息。
* 静态成员函数：void GetState()，按照书店人数:用户总数，书店共享书数量:书籍总数，人均共享数量:人均书籍量的格式打印一条消息，其中人均书籍量只保留整数部分，具体请参考测试说明。

**提示**：可以增加 UserCount ，BookCount 两个静态变量用于记录已有用户数和已有书籍数。

##### 测试说明

平台会对你编写的代码进行测试，比对你输出的数值与实际正确数值，只有所有数据全部计算正确才能通过测试：

测试输入：厉宏富 10 冷欣荣 2 叶文光 0

预期输出：

1. 厉宏富 10 进入
2. 冷欣荣 2 进入
3. 书店人数:2，书店共享书数量:12，人均共享数量:6
4. 厉宏富 10 离开
5. 叶文光 0 进入
6. 书店人数:2，书店共享书数量:2，人均共享数量:1
7. 冷欣荣 2 离开
8. 书店人数:1，书店共享书数量:0，人均共享数量:0
9. 叶文光 0 离开

#### 代码与运行

*#include <string>*

*#include <iostream>*

*using namespace std;*

*/\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/*

*class User {*

*public:*

*string name;*

*int book;*

*static int user\_count, book\_count;*

*static void GetState();*

*User(string name, int book) {*

*this->name = name;*

*this->book = book;*

*User::user\_count++;*

*User::book\_count += book;*

*std::cout << this->name << ' ' << this->book << " 进入" << endl;*

*}*

*~User() {*

*User::user\_count--;*

*User::book\_count -= this->book;*

*std::cout << this->name << ' ' << this->book << " 离开" << endl;*

*}*

*};*

*int User::user\_count = 0, User::book\_count = 0;*

*void User::GetState() {*

*std::cout << "书店人数:" << User::user\_count;*

*std::cout << "，书店共享书数量:" << User::book\_count;*

*std::cout << "，人均共享数量:" << User::book\_count / User::user\_count << endl;*

*}*

*/\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/*

## 3.领悟与反思