

Project 3: 实验报告: 校园卡管理 系统类的编写

实验人: 王翔宇

学号: 15331302

日期: 2016-05-17

院系: 数据科学与计算机学院

专业班级: 软件工程15级教务一班

一、实验目的:

让c++初学者熟悉面向对象的编程思想及类的使用

二、实验环境

VIM – Vi Improved 7.3(2010 Aug 15), Normal version without GUI.

OS X El Captain 10.11.4

Apple LLVM version 7.3.0(clang-703.0.31)

2.1 编程语言&开发环境

ANSI C/C++

开发工具: VIM

2.2 编码规范

要求遵循良好的程序设计风格来设计和编写程序。基本编码规范:

1. 标识符的命名要到达顾名思义的程度;
2. 关键代码提供清晰、准确的注释;
3. 程序版面要求:
 - a) 不同功能块用空行分隔;
 - b) 一般一个语句一行;
 - c) 语句缩进整齐、层次分明。

基本遵循Google style

三、实验内容

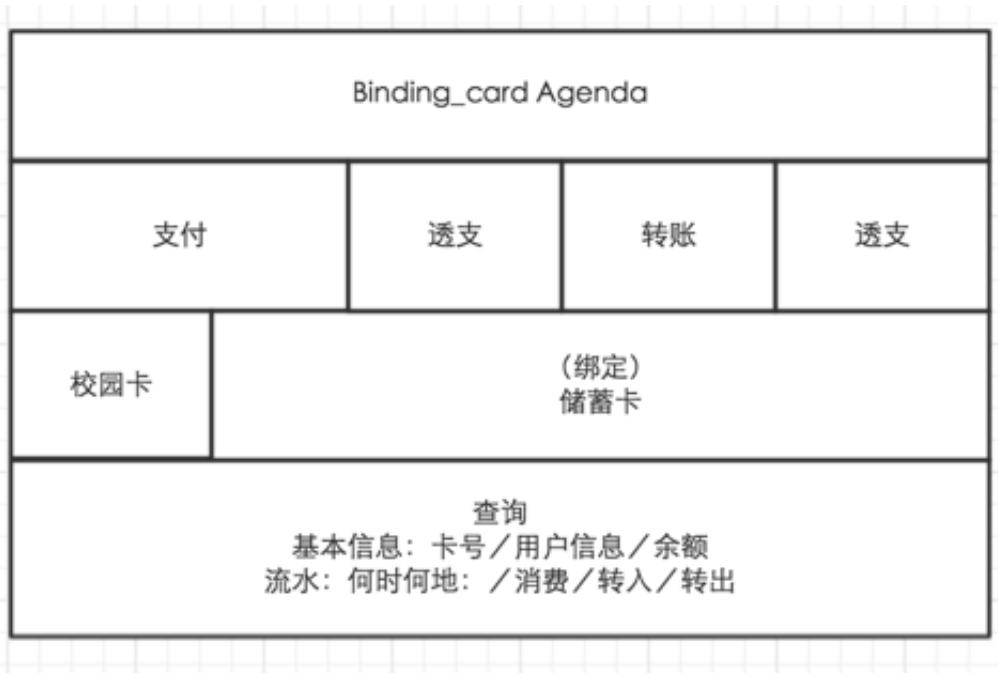
系统设计要求:

- 1.1 编写一个基类 card 类,card 类至少包括两个虚函数,分别是支付函数和查询函数。
- 1.2 编写两个继承 card 类的子类 campus_card 类和 deposit_card 类,分别表示校园卡类和储蓄卡类。两个子类都需要编写继承基类的支付函数和查询函数。查询函数包括查询收支流水记录以及查询该电子卡信息。
- 1.3 编写 campus_card 类和 deposit_card 类的共同子类 binding_card 类,表示绑定卡类。
- 1.4 使用面向对象方法去实现,根据以上对话内容确定更多类的成员函数。通过对对象数组或者文件流输入输出来保存全部电子卡的信息。
- 1.5 设计用户友好的界面,方便用户选择电子卡类型,查看电子卡信息,使用电子卡的功能。

四、分析与设计

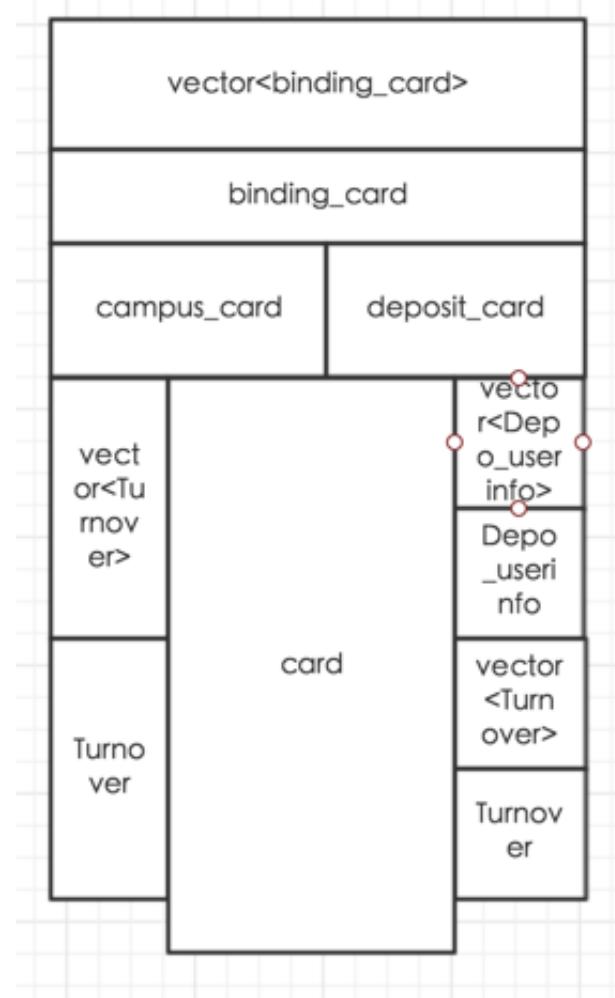
4.1、需求分析:

系统功能图



4.2、类结构设计

类关系图:



4.3、细节设计

```

9
10 --- 7 lines: string getCurrentTime() {-----}
17
18 --- 9 lines: struct BIL {-----}
27
28 --- 66 lines: class Turnover {-----}
94
95 --- 79 lines: class Depo_userinfo {-----}
174
175 --- 5 lines: class card {-----}
180
181 --- 54 lines: class campus_card : public card {-----}
235
236 --- 73 lines: class deposit_card : public card {-----}
309
310 --- 53 lines: class binding_card : public campus_card, public deposit_card {-----}
363
364 --- 73 lines: void readFromFiles(vector<binding_card> &cards) {-----}
437
438 --- 8 lines: void listAll(vector<binding_card> &cards) {-----}
446
447 --- 29 lines: void saveToFile(vector<binding_card> &cds) {-----}
476
477 --- 46 lines: int main() {-----}
    
```

```
28 class Turnover {
29     public:
30         int type; // 0 for income, 1 for payment
31         string time;
32
33         BIL bill;
34
35         string otherCardNo;
36         double ioc; // for other cards' transfer
37
38         double tmpBalance;
39
40     Turnover() {}
41 +--- 6 lines: Turnover(int tp, BIL &bill, double tBalance) {-----}
42 +--- 17 lines: Turnover(int tp, string &other, double ioc, double tBalance) {-----}
43
44         void setTime(string Time) { time = Time; }
45         void setType(int Type) { type = Type; }
46         void setConsumption(int ConsumptionT) { bill.Consumption = ConsumptionT; }
47         void setLocation(string LocationT) { bill.Location = LocationT; }
48
49 +--- 5 lines: void setIOcome(string otherCN, double ioc, double tmpBalance) {-----}
50 +--- 17 lines: void print() {-----}
51
52     };
53 }
```

```
95 class Depo userinfo {
96     public:
97         string cardNo;
98         string releaseDate; // yyyyymmdd
99         string userName;
100        double balance;
101        double overdraftLimit;
102        vector<Turnover> turnovers;
103    };
104    Depo userinfo() { balance = 0; }
105 +--- 7 lines: Depo userinfo(string No, string rld, string usrD, double balance, double overdraftLimit) {---}
106 +--- 7 lines: void setDepo(string No, string rld, string usrD, double balance, double overdraftLimit) {---}
107 +--- 5 lines: void transferToCampus(double money, string stuNo) {---}
108 +--- 5 lines: void income(string &other, double inc) {---}
109 +--- 5 lines: void transferMoney(string &other, double outc) {---}
110
111 +--- 5 lines: void printTurnovers() {---}
112 +--- 3 lines: void printBasisInfo() {---}
113
114 +--- 8 lines: virtual void pay(BIL bill) {---}
115
116 +--- 7 lines: void set(string No, string rld, string usrD, double balance, double overdraftLimit) {---}
117 +--- 3 lines: void setCardNo(string No) {---}
118 +--- 3 lines: void setReleaseDate(string rld) {---}
119 +--- 3 lines: void setUserNames(string usrD) {---}
120 +--- 3 lines: void setBalance(int balance) {---}
121 };
122
```

```
75 class card {
76     public:
77         virtual double pay(BIL bill) = 0;
78         virtual void query(void) = 0; //
79 };
```

```

l81 class campus_card : public card {
l82     public:
l83         double cbalance;
l84         string studentNo; // 8 digits
l85         string stuName;
l86         string school;
l87         vector<Turnover> turnovers;
l88         campus_card() { cbalance = 0; }

l89
l90 +--- 6 lines: void setCamp(string stuNo, string stuN, string sch, double balance) {-----
l96 +--- 5 lines: void input() {-----}
l01 +--- 3 lines: void printStuInfo() {-----}
l04 +--- 5 lines: void getMoneyFromDepo(double money, string otherCardNo) {-----}
l09
l10 +--- 5 lines: void printTurnovers() {-----}
l15 +--- 6 lines: virtual double pay(BIL bill) {-----}
l21 +--- 13 lines: virtual void query(void) {-----}
l34 };

```



```

l81 class campus_card : public card {
l82     public:
l83         double cbalance;
l84         string studentNo; // 8 digits
l85         string stuName;
l86         string school;
l87         vector<Turnover> turnovers;
l88         campus_card() { cbalance = 0; }

l89
l90 +--- 6 lines: void setCamp(string stuNo, string stuN, string sch, double balance) {-----
l96 +--- 5 lines: void input() {-----}
l01 +--- 3 lines: void printStuInfo() {-----}
l04 +--- 5 lines: void getMoneyFromDepo(double money, string otherCardNo) {-----}
l09
l10 +--- 5 lines: void printTurnovers() {-----}
l15 +--- 6 lines: virtual double pay(BIL bill) {-----}
l21 +--- 13 lines: virtual void query(void) {-----}
l34 };

```



```

l0 class binding_card : public campus_card, public deposit_card {
l1     public:
l2         binding_card() {}
l3         vector<Turnover>& CampTurns() { return campus_card::turnovers; }
l4         vector<Depo_userinfo>& DepoCards() { return cards; }
l5 +--- 13 lines: virtual double pay(BIL bill) {-----}
l8 +--- 13 lines: virtual void query(void) {-----}
l1 +--- 21 lines: void transferIn() {-----}
l2 };

```

```
181 class campus_card : public card {  
182     public:  
183         double cbalance;  
184         string studentNo; // 8 digits  
185         string stuName;  
186         string school;  
187         vector<Turnover> turnovers;  
188         campus_card() { cbalance = 0; }  
189  
190     ---- 6 lines: void setCamp(string stuNo, string stuN, string sch, double balance) {----  
196     ---- 5 lines: void input() {-----  
201     ---- 3 lines: void printStuInfo() {-----  
204     ---- 5 lines: void getMoneyFromDepo(double money, string otherCardNo) {-----  
209  
210     ---- 5 lines: void printTurnovers() {-----  
215     ---- 6 lines: virtual double pay(BIL bill) {-----  
221     ---- 13 lines: virtual void query(void) {-----  
234 };
```

五、实验结果

测试包括两个部分，一是文件读写，二是在stdio中输入输出。

文件读写中，所读文件为 `info.db`，所写文件为 `infoTest.db`。

stdio中的输入全部保存在文件中，可直接用 `./a.out < input.txt` 查看输出结果。

说明：

```

MarshalldeMacBook-Pro:proj3-cppcourse Marshall$ ./a.out < input.txt
Testing FILE I/O. Start reading some file.
OPEN file succeeded.
info.db Loaded Successfully.
Testing the functions of the class {binding_card} respectively :

Test for the basic information's input :
Enter the studentNo, studentName, school:
Successfully registered. The initial balance of the campus card is 0, and must be transferred in from the binding deposit_cards.
Enter the new Deposit card's No, releaseDate, userName, balance, overdraftLimit, split by space(s) :
Enter 'n' to exit the Deposit cards Adding, Enter any key to continue Adding : Enter the new Deposit card's No, releaseDate, userName, balance, overdraftLimit, split by space(s) :
Enter 'n' to exit the Deposit cards Adding, Enter any key to continue Adding : Enter 1 to list the infomation of campus_card, 2 for deposit_card :
Enter 1 for the basic information, 2 for turnovers :
0 555555555555555555 20101020 FFather balance : 2e+07 overdraftLimit : 1e+06
1 4545454545454545 21121221 FFFer balance : 2.0202e+07 overdraftLimit : 1010

Test for the payment. just one time.
Input the Consumption & the Location , split by spaces :
error : Campus_card doesn't have enough balance.
Try to Pay it from Deposit_cards ..
Successful. Pay by the card : (No.) 555555555555555555

Now let the Campus_card get some money from the corresponding deposit_cards.

Choose one deposit_card :
0 555555555555555555 20101020 FFather balance : 1.93333e+07 overdraftLimit : 1e+06
1 4545454545454545 21121221 FFFer balance : 2.0202e+07 overdraftLimit : 1010
Choose one card and Enter the conresponding number : The chosen card's balance is : 1.93333e+07
Enter the sum of money : Transfer Completed.

Now let some corresponding deposit_card get income from some other deposit_card
0 555555555555555555 20101020 FFather balance : 1.49899e+07 overdraftLimit : 1e+06
1 4545454545454545 21121221 FFFer balance : 2.0202e+07 overdraftLimit : 1010
Enter the otherCardNo and the sum of money, split by spaces :
Now let some corresponding deposit_card transferMoney() to some other deposit_card
0 555555555555555555 20101020 FFather balance : 1.49899e+07 overdraftLimit : 1e+06
1 4545454545454545 21121221 FFFer balance : 2.0204e+07 overdraftLimit : 1010
Enter the otherCardNo and the sum of money, split by spaces : Enter 1 to list the infomation of campus_card, 2 for deposit_card :
Enter 1 for the basic information, 2 for turnovers :

```

```

tion of campus_card, 2 for deposit_card :
Enter 1 for the basic information, 2 for turnovers :
55555555555555555555555555555555 : FFather :
2016,05,17,08:19:46,Tuesday,+0800 payment : 666666 At : AppStoretempBalance : 1.93333e+0
7
2016,05,17,08:19:46,Tuesday,+0800 Transfer to Campus_card (No.) 15336666Money : 4.34343e
+06tempBalance : 1.49899e+07
2016,05,17,08:19:46,Tuesday,+0800 outcome : 30 , to : 789898898989899tempBalance : 1.498
99e+07

454545454545454545 : FFFer :
2016,05,17,08:19:46,Tuesday,+0800 outcome : 2000 , to : 878787787878778tempBalance : 2.0
204e+07

15330000 Marshall SDCS
balance : 180
TIME_NOT_GIVEN_ONLY_FOR_TEST income : 200 , from : 8888888888888888tempBalance : 200
TIME_NOT_GIVEN_ONLY_FOR_TEST payment : 20 At : AppStoretempBalance : 180
6228481234123412345 : Marshall :

888888888888888888888888 : MARSHALL :
TIME_NOT_GIVEN_ONLY_FOR_TEST payment : 2000 At : USAtempBalance : 8000
TIME_NOT_GIVEN_ONLY_FOR_TEST outcome : 300 , to : 323232323232323232tempBalance : 7700
TIME_NOT_GIVEN_ONLY_FOR_TEST Transfer to Campus_card (No.) 15330000Money : 200tempBalanc
e : 7500
TIME_NOT_GIVEN_ONLY_FOR_TEST income : 2000 , from : 545454545454545454tempBalance : 95
00

15339999 Llahsram SDCS
balance : 495
TIME_NOT_GIVEN_ONLY_FOR_TEST income : 500 , from : 777777777777777777tempBalance : 500
TIME_NOT_GIVEN_ONLY_FOR_TEST payment : 25 At : FoodtempBalance : 475
TIME_NOT_GIVEN_ONLY_FOR_TEST income : 20 , from : 777777777777777777tempBalance : 495
777777777777777777 : Llahsram :
TIME_NOT_GIVEN_ONLY_FOR_TEST Transfer to Campus_card (No.) 15339999Money : 500tempBalanc
e : 49500

15336666 Father SDCS
balance : 4.34343e+06
2016,05,17,08:19:46,Tuesday,+0800 income : 4.34343e+06 , from : 5555555555555555tempBa
lance : 4.34343e+06
55555555555555555555555555555555 : FFather :
2016,05,17,08:19:46,Tuesday,+0800 payment : 666666 At : AppStoretempBalance : 1.93333e+0
7

```

```
15336666 Father SDCS
balance : 4.34343e+06
2016,05,17,08:19:46,Tuesday,+0800 income : 4.34343e+06 , from : 5555555555555555tempBa
lance : 4.34343e+06
5555555555555555 : FFather :
2016,05,17,08:19:46,Tuesday,+0800 payment : 666666 At : AppStoretempBalance : 1.93333e+0
7
2016,05,17,08:19:46,Tuesday,+0800 Transfer to Campus_card (No.) 15336666Money : 4.34343e
+06tempBalance : 1.49899e+07
2016,05,17,08:19:46,Tuesday,+0800 outcome : 30 , to : 789898898989899tempBalance : 1.498
99e+07

4545454545454545 : FFFer :
2016,05,17,08:19:46,Tuesday,+0800 outcome : 2000 , to : 878787787878778tempBalance : 2.0
204e+07
```

具体的测试内容在terminal中已有详细说明，此处不再赘述。

六、设计心得

更加了解了agenda的编写模式。这已经是我们在本门课程完成的第四个agenda，这增加了我的经验。

体会：设计是关键，与技术同样重要。