



# 实验报告

实验人：黄玟瑜

学号：19335074

日期：2020 年 6 月 9 日

院（系）：数据科学与计算机学院

专业（班级）： 计算机类行政 3 班

实验题目： 校园卡管理项目设计

## 1. 实验目的

设计一个校园卡管理系统，包含以下核心功能：

1. 查看卡主和校园卡信息
2. 充值
3. 消费
4. 查余
5. 查询个人消费记录
6. 充值记录

## 二. 实验环境

本实验可基于Visual Studio Code等平台开发，参考主流的编码规范，如[Google C++Style Guide](#)（中文版）

### 2.1 编程语言和开发工具

编程语言： ANSI C/C++

开发工具： Visual Studio Code、Dev C++

### 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。

## 3. 实验内容

```
4. #include <cstdlib>
5. #include <cstring>
6. #include <deque>
7. #include <iomanip>
8. #include <iostream>
9. #include <string>
10. #include "Student.cpp"
11. #include "Student.hpp"
12. #include "Card.cpp"
13. #include "Card.hpp"
```

```
14.
15. #define MAX 10000
16. using namespace std;
17.
18. void ManagerFunction();
19. void StudentFunction();
20.
21. Student *students[MAX] = {NULL};
22. Card *cards[MAX] = {NULL};
23.
24. int main()
25. {
26.     int select;
27.     do
28.     {
29.         cout << "欢迎使用校园卡管理系统！请选择您的身份（输入序号）："
30.         << endl;
31.         cout << "1: 校园卡管理员" << endl;
32.         cout << "2: 学生用户" << endl;
33.         cout << "3: 退出系统" << endl;
34.         cin >> select;
35.         switch (select)
36.         {
37.             case 1:
38.                 ManagerFunction();
39.                 break;
40.             case 2:
41.                 StudentFunction();
42.                 break;
43.             case 3:
44.                 break;
45.             default:
46.                 cout << "抱歉，该身份无效！" << endl;
47.                 break;
48.         }
49.
50.
51.         cout << endl;
52.     } while (select!=3);
53.
54.     cout << endl;
55.     cout << "已退出系统！" << endl;
```

```

56.     for (int i = 0; i < MAX;i++){
57.         if(students[i]!=NULL)
58.             delete students[i];
59.         if(cards[i]!=NULL)
60.             delete cards[i];
61.     }
62.     return 0;
63. }

```

根据校园卡或管理员身份设计功能，其中校园卡管理员的权利包括：

```

void ManagerFunction()
{
    void CreateStudent();
    void CreateCard();
    void Transfer();
    void DeleteCard();
    void DeleteStudent();
    .....
}

```

学生类的权利包括：

```

void Check();
void Activate();
void Charge();
void ChargeCard();
void ChargeWater();
void Consume();
void Loss();
.....

```

## 4. 分析与设计

简要描述程序设计的过程，包括设计思路，设计要点及特色；程序的不足与改进等。要求画出程序的简单流程图。

### 4.1 、需求分析：

#### 一、校园卡管理员

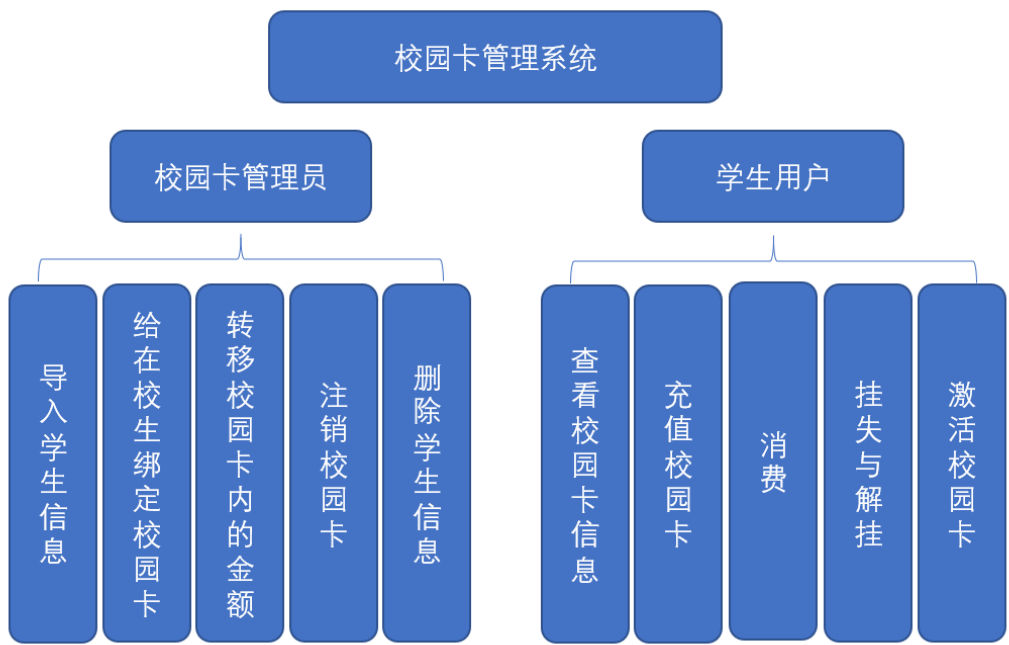
1. 导入学生信息：向系统中导入学生信息（名字、学院、年级、班级、专业、学号）
2. 给在校生绑定校园卡（卡号，余额、充值记录、消费记录、校园卡状态、水控服务）
3. 转移校园卡内的金额：学生本人申请办理新卡时，可将原账户的余额转移到新卡中
4. 注销校园卡：校园卡一经注销后无法使用。丢失校园卡后要重新办理的学生，需将原来绑定的校园卡注销后再办理新卡。
5. 删除学生信息：当学生毕业或退学，管理员可将系统中的学生信息删除。

#### 二、学生用户

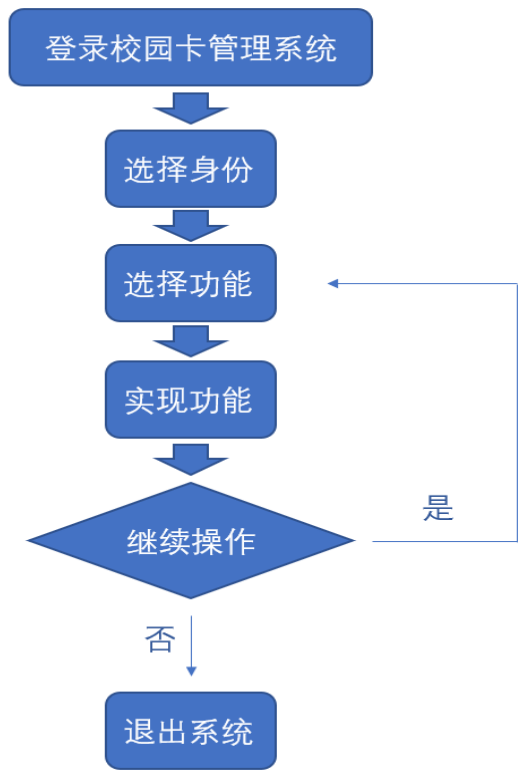
1. 查看校园卡信息（卡号，余额、充值记录、消费记录、校园卡状态、水控服务）
2. 激活校园卡：设置密码
3. 充值校园卡：校园卡充值限额为 100~500 元

4. 消费：学生可到食堂超市（第一饭堂、第二饭堂、第三饭堂、第四饭堂、行政楼餐厅、清真食堂、教育超市、快乐多超市）进行消费。可通过热水账户的金额使用热水（洗澡）。
5. 挂失与解挂：校园卡丢失后，学生可进行挂失，挂失后的校园卡无法使用。若校园卡被找回，可进行解挂，解挂后校园卡可以继续使用。

系统功能图



流程图



## 4.2、 类结构设计

### 1.学生类

```
1. class Student{
2.     const string Name;
3.     int ID,Department, Major, Grade, Class;
4.     Card *card;
5.
6. public:
7.     Student(string Name_, int ID_,int Department_, int Major_, int Grade_, int Class_);
8.     void setCard(Card *);
9.     void Check();
10.    void Activate();
11.    void Charge();
12.    void ChargeCard();
13.    void ChargeWater();
14.    void Consume();
15.    void Loss();
16.
17.    string getName()const { return Name; }
18.    int getID()const { return ID; }
19.    int getDepartment()const { return Department; }
20.    int getMajor()const { return Major; }
21.    int getGrade()const { return Grade; }
22.    int getClass()const { return Class; }
23.
24.    Card *getCard() const { return card; }
25.};
```

### 2.校园卡类

```
class Card{
    const int Number;
    string Password;
    double Value, WaterAccount;
    state State;
    Student *student;
    deque<ChargeRecord*> charge_record;
    deque<ConsumeRecord*> consume_record;
    deque<WaterRecord*> water_record;
public:
    Card(int Num, Student *);
    bool isChecked();
```

```

    void add(double amount);
    void minus(double amount);
    void addWater(double amount);
    void minusWater(double amount);
    void setState(state);
    void setPassword(string);

    void recordCharge(ChargeRecord *);
    void recordConsume(ConsumeRecord *);
    void recordWater(WaterRecord *);
    void printCharge() const;
    void printConsume() const;
    void printWater() const;

    int getNum()const { return Number; }
    double getValue()const { return Value; }
    double getWaterAccount()const { return WaterAccount; }
    int getState()const { return State; }
};

```

### 3.记录类及其派生出的充值记录类、消费记录类、热水使用记录类

```

class Record{
    int Month, Day, Hour, Minute, Second;
    double Amount;

public:
    Record(double amount=0);
    void set(double amount) { Amount = amount; }
    void Print_Record()const;
};

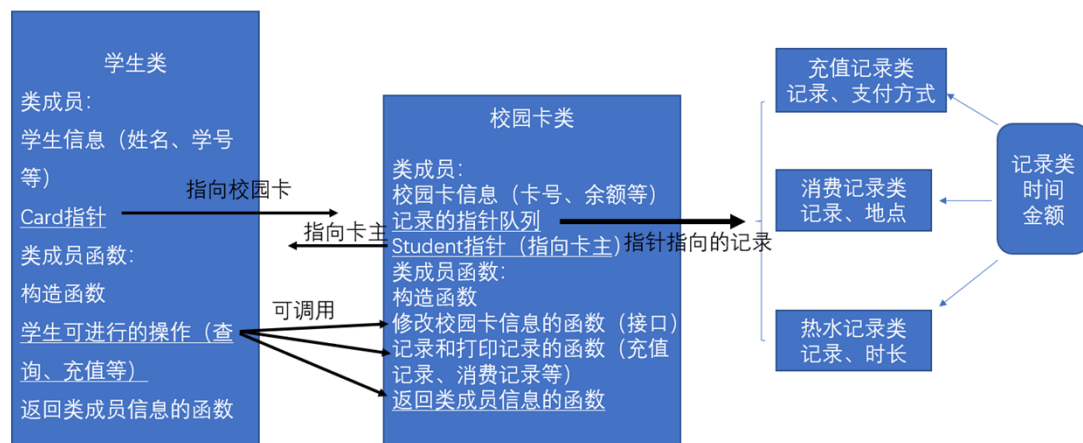
class ChargeRecord:public Record{
    int Method;
public:
    ChargeRecord(double amount );
    void Print_ChargeRecord() const;
};

class ConsumeRecord:public Record{
    int Place;
public:
    ConsumeRecord(double amount);
    void Print_ConsumeRecord() const;
};

```

```
class WaterRecord:public Record{
    int Time;
public:
    WaterRecord(int );
    void Print_WaterRecord() const;
};
```

类关系图:



### 4.3、 细节设计

1.

```
int select;
do
{
    ...
    cout << "3: 退出" << endl;
    cin >> select;

    switch (select)
    {
        case ...
        case 3:
            break;
        default:
            cout << "抱歉, 该身份无效!" << endl;
            break;
    }
    cout << endl;
} while (select!=3);
```

每一次选择都保证选项有效，且只有用户选择退出时才退出。

2.

```
cout << "\n 请输入学生学号: " << endl;
cin >> ID;
while (ID < 0 || ID >= MAX)
{
    cout << "\n 学号格式错误! 请输入正确的学号: " << endl;
    cin >> ID;
}
if (students[ID] != NULL)
{
    cout << "\n 该学生信息已存在! 请重新输入: " << endl;
    return;
}
```

```
cout << "\n 请输入转出账户卡号: " << endl;
cin >> from;
while (from < 0 || from >= MAX)
{
    cout << "\n 卡号格式有误! 请重新输入: " << endl;
    cin >> from;
}
while (cards[from] == NULL)
{
    cout << "\n 该账户不存在! 请重新输入: " << endl;
    cin >> from;
}
```

每次输入学号或卡号都会检查学号和卡号的正确性，若学号或卡号的格式正确，将会判断该学生信息或校园卡是否存在系统中。

3.

```
...
case 2: if(students[ID]->getCard()->getState()==LOST){
    cout << endl;
    cout << "此卡处于挂失状态无法充值! " << endl;
}
else
    students[ID]->Charge();
break;
case 3: if(students[ID]->getCard()->getState()==LOST){
    cout << endl;
```



```

        cout << "此卡处于挂失状态无法进行消费！" << endl;
    }
    else
        students[ID]->Consume();
    break;

```

...

当校园卡处于挂失状态时无法进行充值和消费。

4.

```

    if (students[ID]->getCard()->getState() == OFF)
    {
        cout << "您的校园卡尚未激活，是否激活校园卡？\n 输入 1 激活，输入 0 退出：" << endl;
        cin >> select;
        if (select)
            students[ID]->Activate();
        else
            return;
    }

```

...

```

        case 4:if (students[ID]->getCard()->isChecked())
            students[ID]->Loss();

```

...

若学生要激活校园卡、挂失或解挂，将要验证密码。

5.

```

bool Card::isChecked(){
    string temp;
    for (int i = 9; i >= 0;i--){
        cout << endl;
        cout << "请输入校园卡密码：" << endl;
        cin >> temp;

        if(temp==Password){
            cout << "密码正确！" << endl;
            return true;
        }
        else
            cout << "密码错误！还剩 " << i << " 次机会" << endl;
    }
    cout << "机会已用完！验证失败。" << endl;
    return false;
}

```

验证密码共有 10 次机会，若密码正确则返回 true，若 10 次输入都密码错误将返回 false，同时提示验证失败。

6.

```
...
case 1:
    cout << endl;
    cout << "当前余额为: " << card->getValue() << " 元" << endl;
    cout << "请输入消费金额: " << endl;
    in >> amount;

    if(amount > card->getValue())
        cout << "校园卡余额不足!" << endl;
    else{
        card->minus(amount);
        card->recordConsume(new ConsumeRecord(amount));

        cout << endl;
        cout << "成功消费 " << amount << " 元!" << endl;
        cout << "当前余额: " << card->getValue() << " 元" << endl;
    }

    break;
```

```
Record::Record(double amount):Amount(amount){
    cout << "\n 请输入日期（月、日，用空格隔开）: " << endl;
    cin >> Month >> Day;
    cout << "\n 请输入当前时间（时、分、秒，用空格隔开）: " << endl;
    cin >> Hour >> Minute >> Second;
}
ChargeRecord::ChargeRecord(double amount):Record(amount){
    cout << endl;
    cout << "请选择支付方式: " << endl;
    for (int i = 0; i < 3;i++){
        cout << i + 1 << ": " << method[i] << endl;
    }
    cin >> Method;

    cout << endl;
    cout << "成功充值 " << amount << " 元!" << endl;
}
```

学生每次进行充值、消费、使用热水都会进行记录，记录金额、日期（月、日）、时间（时、分、秒）、支付方式或消费地点等信息。

## 5、实验结果

样例说明：

```
1
1          //新生入学，校园卡管理员为新生录入学生信息
0123      //学号
张三      //姓名
1          //选择院系，数据科学与计算机学院
1          //选择专业，计算机系
2019      //年级，即入学年份
3          //班级
2          //绑定校园卡，卡号随机且具有唯一性
0123
6          //退出
2          //登入学生系统
0123      //输入学号
1          //激活校园卡
000000    //输入初始密码
123456    //输入新密码
2          //充值校园卡
1          //余额充值
600        //输入充值金额，大于限额（500 元）时提示充值失败
100        //输入正确金额，充值成功
12 25     //输入充值的日期（月、日）
12 12 12  //时间（时、分、秒）
1          //选择支付方式。生成充值记录。
3          //退出
3          //消费
1          //饭堂超市消费
101        //消费金额大于余额时，消费失败
1          //重新选择饭堂超市消费
```

```
10      //输入消费金额
12 25    //输入信息生成消费记录
12 12 13
1
3        //退出
1        //查询
3        //查看充值记录
4        //查看消费记录
6        //退出查询
5        //退出学生系统
3        //退出系统，关闭程序
```

实验结果：

d:\C++ practise\practise\vscode\PROJECT1\main.exe

欢迎使用校园卡管理系统！请选择您的身份（输入序号）：

- 1: 校园卡管理员
- 2: 学生用户
- 3: 退出系统

1

管理员您好！

- 1: 导入学生信息
- 2: 创建并绑定校园卡
- 3: 校园卡余额转账
- 4: 注销校园卡
- 5: 删除学生信息
- 6: 退出

请选择操作：

1

请输入学生学号：

0123

请输入学生姓名：

张三

- 1: 数据科学与计算机学院
- 2: 数学学院
- 3: 中山医学院
- 4: 管理学院
- 5: 岭南学院
- 6: 外国语学院
- 7: 马克思主义学院

请选择学生院系：

1

- 1: 计算机系
- 2: 数学系
- 3: 医学
- 4: 工商管理
- 5: 经济学
- 6: 英语系
- 7: 哲学

请选择学生专业：

1

请输入学生年级（入学年份）：

2019

请输入学生班级：

3

成功导入学生信息！

学号：0123

姓名：张三

成功导入学生信息！  
学号：0123      姓名：张三  
院系专业：数据科学与计算机学院计算机系  
年级班级：2019级3班

1: 导入学生信息  
2: 创建并绑定校园卡  
3: 校园卡余额转账  
4: 注销校园卡  
5: 删除学生信息  
6: 退出  
请选择操作：

2

请输入学生学号：  
0123

学号为 0123 的学生成功绑定校园卡 0041

1: 导入学生信息  
2: 创建并绑定校园卡  
3: 校园卡余额转账  
4: 注销校园卡  
5: 删除学生信息  
6: 退出  
请选择操作：

6

欢迎使用校园卡管理系统！请选择您的身份（输入序号）：

1: 校园卡管理员  
2: 学生用户  
3: 退出系统

2

同学您好！请输入您的学号：

0123

您的校园卡尚未激活，是否激活校园卡？

输入1激活，输入0退出：

1

请输入校园卡密码：

000000

密码正确！

请输入新密码（6位数可包含数字、字母、下划线）：

123456

成功激活校园卡！

卡号：0041

状态：已激活

状态：已激活

- 1: 查看校园卡信息
- 2: 充值校园卡
- 3: 消费
- 4: 挂失与解挂
- 5: 退出

请选择操作：

2

- 1: 余额充值
- 2: 热水账户充值
- 3: 退出

请选择：

1

请输入充值金额（100~500）：

600

充值限额为100~500，请重新输入：

100

请输入日期（月、日，用空格隔开）：

12 25

请输入当前时间（时、分、秒，用空格隔开）：

12 12 12

请选择支付方式：

- 1: 现金支付
- 2: 微信支付
- 3: 支付宝支付

1

成功充值 100 元！

卡号：0041

余额：100 元

- 1: 余额充值
- 2: 热水账户充值
- 3: 退出

请选择：

3

- 1: 查看校园卡信息
- 2: 充值校园卡
- 3: 消费
- 4: 挂失与解挂
- 5: 退出

请选择操作：

101

校园卡余额不足！

1: 饭堂超市消费

2: 热水消费

3: 退出

请选择:

1

当前余额为: 100 元

请输入消费金额: (限额为500元)

10

请输入日期(月、日, 用空格隔开):

12 25

请输入当前时间(时、分、秒, 用空格隔开):

12 12 13

请选择消费地点:

1: 第一饭堂

2: 第二饭堂

3: 第三饭堂

4: 第四饭堂

5: 清真食堂

6: 行政楼餐厅

7: 教育超市

8: 快乐多超市

1

成功消费 10 元!

当前余额: 90 元

1: 饭堂超市消费

2: 热水消费

3: 退出

请选择:

3

1: 查看校园卡信息

2: 充值校园卡

3: 消费

4: 挂失与解挂

5: 退出

请选择操作:

1

-----卡片信息-----

卡号: 0041

状态: 已激活



3: 充值记录  
4: 消费记录  
5: 热水使用记录  
6: 退出  
请选择:  
3

以下是该校园卡的近10条充值记录:

金额	时间	支付方式
100	12月25日12时12分12秒	现金支付

1: 查询余额  
2: 热水账户  
3: 充值记录  
4: 消费记录  
5: 热水使用记录  
6: 退出  
请选择:

4

以下是该校园卡的近10条消费记录:

金额	时间	消费地点
10	12月25日12时12分13秒	第一饭堂

1: 查询余额  
2: 热水账户  
3: 充值记录  
4: 消费记录  
5: 热水使用记录  
6: 退出  
请选择:

6

1: 查看校园卡信息  
2: 充值校园卡  
3: 消费  
4: 挂失与解挂  
5: 退出  
请选择操作:

5

欢迎使用校园卡管理系统! 请选择您的身份(输入序号):

1: 校园卡管理员  
2: 学生用户  
3: 退出系统

3

已退出系统!  
请按任意键继续. . .

## 6、设计心得

通过分析需求和设计层次我设计了一个能满足需求的校园卡管理系统，还增加了热水账户功能，使校园卡同时具备热水卡的功能。设计了一些模块来确保学号、卡号的唯一性，设计了充值或消费时超出限额时的提醒功能。同时针对客户输入无效选项时设置了提醒。

此项目中的校园卡类、学生类、记录类采用的是堆空间存储，并且在程序使用完毕时释放。

但仍有不足的地方，如：

1. 采用指针数组的方法来存储每张校园卡的地址，卡号为数组下标。这种方法虽然确保了卡号的唯一性，可对于用户来说输入卡号 41 和 0041 效果是一样的。而这并不能很好的确保卡号的规范。学生类也存在这个问题。
2. 设置校园卡密码时虽规定 6 位密码且仅能使用字母、数字、下划线，但由于采用了 string 类来存储密码，因此不能保证密码的规范性。
3. 充值记录类能显示充值记录，但充值记录没有校园卡余额充值或热水账户充值的说明。