****

面向对象程序设计上机作业

**作业名称：**  一卡通乘车系统

**作业作者：**  黄梓枫

**学 号：**  14130130295

**所在院系：** 软件学院

一、类设计

class Card {

friend class System;

friend class StudentCard;

friend class LimitedCard;

friend class TeacherCard;

friend class TemporaryCard;

protected:

int record\_time;

int money;

int type;

int num;

std::string name;

enum CARD\_TYPE { STUDENT\_CARD, TEACHER\_CARD, LIMITED\_CARD, TEMPORARY\_CARD };

//TEMPORARY\_CARD

time\_t START\_TIME;

int TOTAL\_TIME;

bool STATE;

enum { ACTIVE, FROZEN }; // CARD\_STATE

public:

Card(const std::string NAME = "NULL", int MONEY = 0);

virtual ~Card() {}

virtual bool record() {}

virtual void show() const;

void write(std::ofstream &fout) const;

bool operator < (const Card& card) const;

void charge(int MONEY);

};

class LimitedCard : public Card {

friend class System;

private:

LimitedCard(const std::string NAME = "NULL", int MONEY = 0);

LimitedCard(const Card& card);

virtual bool record();

virtual void show() const;

};

class TeacherCard : public Card {

friend class System;

private:

TeacherCard(const std::string NAME = "NULL", int MONEY = 0);

TeacherCard(const Card& card);

virtual bool record();

virtual void show() const;

};

class StudentCard : public Card {

friend class System;

private:

StudentCard(const std::string NAME = "NULL", int MONEY = 0);

StudentCard(const Card& card);

virtual bool record();

virtual void show() const;

};

class Bus {

friend class System;

private:

int num; //车的编号

std::string VIN;//vehicle identification number //车牌号

std::string MODEL; //型号

int MAX\_NUMBER\_OF\_PEOPLE; //核载人数

int number\_of\_people; //实载人数

std::string NAME\_OF\_DRIVER; //司机姓名

time\_t START\_TIME, END\_TIME; //起止时间

private:

Bus(std::string \_VIN, std::string \_MODEL, int \_MAX,

std::string \_NAME, time\_t \_START\_TIME, time\_t \_END\_TIME);

Bus(int \_NUM, std::string \_VIN, std::string \_MODEL, int \_MAX, int

\_NUMBER\_OF\_PEOPLE, std::string \_NAME, time\_t \_START\_TIME, time\_t \_END\_TIME) :

num(\_NUM), VIN(\_VIN), MODEL(\_MODEL), MAX\_NUMBER\_OF\_PEOPLE(\_MAX),

number\_of\_people(\_NUMBER\_OF\_PEOPLE),NAME\_OF\_DRIVER(\_NAME),

START\_TIME(\_START\_TIME), END\_TIME(\_START\_TIME) {}

bool isFull() { return number\_of\_people >= MAX\_NUMBER\_OF\_PEOPLE; }

bool add();

void show() const;

void write(std::ofstream &fout) const;

public:

bool operator < (const Bus& bus) const;

};

class System {//单例模式

private:

static System \*instance;

std::set<Card> CARD\_SET;

std::set<Bus> BUS\_SET;

std::set<Card>::iterator CARD\_IT;

std::set<Bus> ::iterator BUS\_IT;

std::ifstream fin;

std::ofstream fout;

enum CARD\_TYPE { STUDENT\_CARD, TEACHER\_CARD, LIMITED\_CARD, TEMPORARY\_CARD };

System() { read(); }

public:

~System() { write(); clearSet(); }

static System\* getInstance() {

if (instance == NULL)

instance = new System();

return instance;

}

TeacherCard\* addTeacherCard(const std::string NAME, int MONEY);

LimitedCard\* addLimitedCard(const std::string NAME, int MONEY);

StudentCard\* addStudentCard(const std::string NAME, int MONEY);

TemporaryCard\* addTemporaryCard(const std::string NAME, int MONEY,

int TOTAL\_TIME);

Bus\* addBus(std::string \_VIN, std::string \_MODEL, int \_MAX, std::string \_NAME);

bool deleteCard(Card &card);

bool deleteBus(Bus &bus);

bool chargeCard(Card& card, int money);

int showCardNumber();

int showBusNumber();

void showAllCardInfo();

void showAllBusInfo();

bool record(Card &card);

void reactiveTemporaryCard(TemporaryCard &tcard, int total\_time);

const Card\* findCardByName(std::string NAME);

void write();

void read();

void clearSet();

};

System类是Card和Bus类的友元类，负责管理Card和Bus的添加(addXXXXCard/addBus)、删除(deleteCard/Bus)、查询(findCardByName)、激活(reactiveTemporaryCard)、磁盘读写(read/write)、显示全部信息(showAllCard/BusInfo)、打卡(record)、充值(chargeCard)等操作。Card和Bus分别保存在System的CARD\_SET和BUS\_SET中，使用对SET的操作(insert/erase)以及迭代器遍历即可实现信息的保存。文件读写(read/write)使用的是fstream，实例化fout和fin就可以在bus.txt和card.txt中保存和读取信息。System类采用的是单例模式，使用getInstance()方法获取对象，并从文件中读取信息。

Card类是一卡通基类，由它派生出TeacherCard、StudentCard、LimitedCard三个子类。它有打卡(record)、显示信息(show)、充值(charge)和写入(write)等操作。

Bus类是公交车类，它由System的addBus分配开始时间为当前时间，公交车运行时间为一个小时，承载量为System分配。它有判断是否客满(isFull)、增加乘客(add)、显示信息(show)等操作。

二、操作详解

1.获取System实例

System \*system = System::getInstance();

2.添加新卡

Card \*card = system->addStudentCard("name", money);

3.添加新公交车

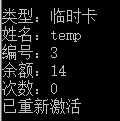
Bus \*bus = system->addBus("VIN", "MODEL", MAX\_NUMBER\_OF\_PEOPLE, "NAME");

4.按名字查找

string name = "temp";

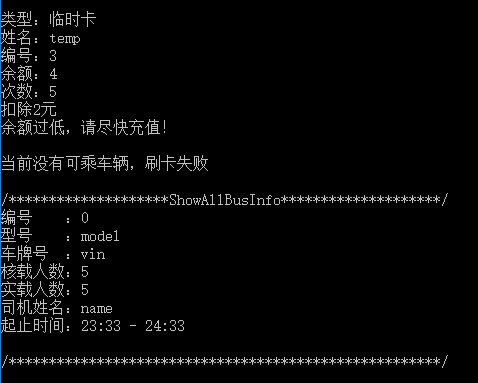
TemporaryCard\* tcard = (TemporaryCard\*)system->findCardByName(name);

5.重新激活临时卡



system->reactiveTemporaryCard(\*tcard, time);

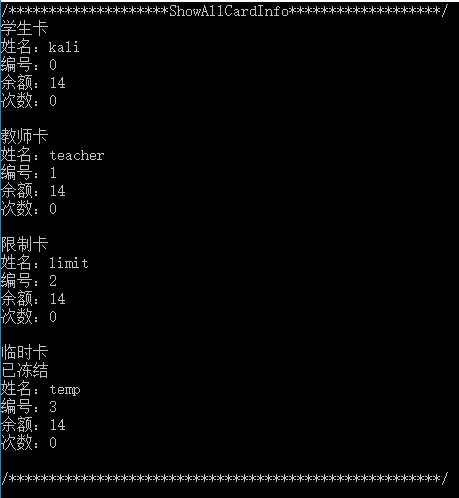
6.打卡乘车



system->record(\*tcard);

7.显示所有信息

system->showAllCardInfo();



system->showAllBusInfo();

8.一卡通充值

system->chargeCard(\*card, money);

9.删除一卡通

system->deleteCard(\*card);

10.删除公交车

system->deleteBus(\*bus);

11.保存数据

system->write();

12.析构系统

delete system;

三、结果分析

基本能解决所提出的问题，类封装和继承、友元、多态、设计模式、STL等所学的知识都能较好地融入该作业中。