//Benzin.cpp

#include "benzin.h"

#include "File.h"

benzin::benzin()

{

rashodt = 0;

}

int benzin::get\_rashodt()

{

return rashodt;

}

void benzin::settingsost()

{

cout << "изменить:\n1)Марку-\n2)Состояние-\n3)Количество мест-\n4)Гос. номер-\n5)Маршрут-\n6)Кузов-\n7)Расход бензина" << endl;

switch (\_getch())

{

case '1':

cout << "";

mark = letters();

break;

case '2':

cout << "";

sost = letters();

break;

case '3':

cout << "";

kmest = number();

break;

case '4':

cout << "Введите в формате 1111АА-1";

set\_gosnomer();

break;

case '5':

cout << "";

marshrut = letters();

break;

case '6':

cout << "";

kuzov = letters();

break;

case '7':

cout << "";

rashodt = number();

break;

defaut:

cout << "" << endl;

break;

}

}

ostream& operator << (ostream& out, benzin& obj)

{

out << setiosflags(ios::left)

<< setw(15) << obj.get\_mark()

<< setw(15) << obj.get\_sost()

<< setw(15) << obj.get\_kmest()

<< setw(14) << obj.get\_gosnomer()

<< setw(20) << obj.get\_marshrut()

<< setw(20) << obj.get\_kuzov()

<< setw(20) << obj.get\_rashodt();

cout << endl;

return out;

};

istream& operator >> (istream& in, benzin& obj)

{

cout << "марка-";

obj.mark = letters();

cout << "Состояние-";

obj.sost = letters();

cout << "Кол-во мест-";

obj.kmest = number();

cout << "Гос. номер-"<< endl;

cout << "Введите в формате 1111АА-1";

obj.set\_gosnomer();

cout << "Маршрут-";

obj.marshrut = letters();

cout << "Кузов-";

obj.kuzov = letters();

cout << "Расход топлива-";

obj.rashodt = number();

return in;

};

void benzin::shap()

{

cout << setiosflags(ios::left)

<< setw(15) << "Марка"

<< setw(15) << "Состояние"

<< setw(15) << "Кол-во мест"

<< setw(14) << "Гос. номер"

<< setw(20) << "Маршрут"

<< setw(20) << "Кузов"

<< setw(20) << "Расход топлива";

cout << endl;

}

bool benzin::operator == (benzin& obj)

{

if ((obj.mark == this->mark || obj.mark == "n") &&

(obj.sost == this->sost || obj.sost == "n") &&

(obj.kmest == this->kmest || obj.kmest == 1) &&

(obj.gosnomer == this->gosnomer || obj.gosnomer == "n") &&

(obj.marshrut == this->marshrut || obj.marshrut == "n") &&

(obj.kuzov == this->kuzov || obj.kuzov == "n") &&

(obj.rashodt == this->rashodt || obj.rashodt == 1))

return true;

else

return false;

}

ifstream& operator >> (ifstream& vvod, benzin& obj)

{

vvod >> obj.mark >> obj.sost >> obj.kmest >> obj.gosnomer >> obj.marshrut >> obj.kuzov >> obj.rashodt;

return vvod;

}

ofstream& operator << (ofstream& vivod, benzin& obj)

{

vivod << obj.mark << ' ' << obj.sost << ' ' << obj.kmest << ' ' << obj.gosnomer << ' ' << obj.marshrut << ' ' << obj.kuzov << ' ' << obj.rashodt << endl;;

return vivod;

}

//elecrobus.cpp

#include "elecrobus.h"

#include "File.h"

elecrobus::elecrobus()

{

rashode = 0;

}

int elecrobus::get\_rashode()

{

return rashode;

}

void elecrobus::settingsost()

{

cout << "изменить:\n1)Марку-\n2)Состояние-\n3)Количество мест-\n4)Гос. номер-\n5)Маршрут-\n6)Кузов-\n7)Расход електричества" << endl;

switch (\_getch())

{

case '1':

cout << "";

mark = letters();

break;

case '2':

cout << "";

sost = letters();

break;

case '3':

cout << "";

kmest = number();

break;

case '4':

cout << "Введите в формате 1111АА-1";

set\_gosnomer();

break;

case '5':

cout << "";

marshrut = letters();

break;

case '6':

cout << "";

kuzov = letters();

break;

case '7':

cout << "";

rashode = number();

break;

defaut:

cout << "" << endl;

break;

}

}

ostream& operator << (ostream& out, elecrobus& obj)

{

out << setiosflags(ios::left)

<< setw(15) << obj.get\_mark()

<< setw(15) << obj.get\_sost()

<< setw(15) << obj.get\_kmest()

<< setw(14) << obj.get\_gosnomer()

<< setw(20) << obj.get\_marshrut()

<< setw(20) << obj.get\_kuzov()

<< setw(20) << obj.get\_rashode();

cout << endl;

return out;

};

istream& operator >> (istream& in, elecrobus& obj)

{

cout << "марка-";

obj.mark = letters();

cout << "Состояние-";

obj.sost = letters();

cout << "Кол-во мест-";

obj.kmest = number();

cout << "Гос. номер-" << endl;

cout << "Введите в формате 1111АА-1";

obj.set\_gosnomer();

cout << "Маршрут-";

obj.marshrut = letters();

cout << "Кузов-";

obj.kuzov = letters();

cout << "Расход электричества-";

obj.rashode = number();

return in;

};

void elecrobus::shap()

{

cout << setiosflags(ios::left)

<< setw(15) << "Марка"

<< setw(15) << "Состояние"

<< setw(15) << "Кол-во мест"

<< setw(14) << "Гос. номер"

<< setw(20) << "Маршрут"

<< setw(20) << "Кузов"

<< setw(20) << "Расход электричества";

cout << endl;

}

bool elecrobus::operator == (elecrobus& obj)

{

if ((obj.mark == this->mark || obj.mark == "n") &&

(obj.sost == this->sost || obj.sost == "n") &&

(obj.kmest == this->kmest || obj.kmest == 1) &&

(obj.gosnomer == this->gosnomer || obj.gosnomer == "n") &&

(obj.marshrut == this->marshrut || obj.marshrut == "n") &&

(obj.kuzov == this->kuzov || obj.kuzov == "n") &&

(obj.rashode == this->rashode || obj.rashode == 1))

return true;

else

return false;

}

ifstream& operator >> (ifstream& vvod, elecrobus& obj)

{

vvod >> obj.mark >> obj.sost >> obj.kmest >> obj.gosnomer >> obj.marshrut >> obj.kuzov >> obj.rashode;

return vvod;

}

ofstream& operator << (ofstream& vivod, elecrobus& obj)

{

vivod << obj.mark << ' ' << obj.sost << ' ' << obj.kmest << ' ' << obj.gosnomer << ' ' << obj.marshrut << ' ' << obj.kuzov << ' ' << obj.rashode << endl;;

return vivod;

}

//Exception.cpp

#pragma once

#include "Exception.h"

Exception::Exception(const Exception& q)

{

numer = q.numer;

strochka = q.strochka;

}

Exception::Exception(int \_numer, const string w)

{

strochka = w;

numer = \_numer;

}

Exception::~Exception()

{

}

void Exception::pokaz()

{

cout << strochka;

cout << endl;

}

int number()

{

int t = 0;

bool f;

do

{

try

{

while (!(cin >> t))

{

cin.clear();

cout << "ERROR\nНеправильный ввод\nВведите число\n";

string c;

cin >> c;

}

f = 1;

if (t < 0)

throw Exception(1, "\nВведите число больше нуля");

}

catch (Exception& obj)

{

f = 0;

obj.pokaz();

}

} while (!f);

return t;

}

string letters()

{

int ko;

rewind(stdin);

string str;

do

{

try

{

ko = 1;

cin >> str;

for (int i = 0; str[i]; i++)

{

if ((str[i] < 'A' || str[i] > 'z') && (str[i] != '-'))

throw Exception(3, "Переключитеь на англискую клавиатуру/Вводите буквы");

}

}

catch (Exception& e)

{

ko = 0;

e.pokaz();

rewind(stdin);

}

} while (!ko);

return(str);

}

//main.cpp

#include"Shablon.cpp"

#include "benzin.h"

#include "turist.h"

#include "pasazhir.h"

#include "Auto.h"

#include "elecrobus.h"

#include "File.h"

template<class T>

void menupolz(shablon <T>& obj)

{

obj.ProsmFile();

while (1)

{

T tmp;

system("cls");

cout << "1)Найти\n3)Удалить\n2)Просмотреть" << endl;

switch (\_getch())

{

case '1':

system("cls");

obj.poisk(tmp);

break;

case'2':

system("cls");

obj.out();

break;

}

}

}

template<class T>

void menuadmin(shablon <T>& obj)

{

obj.ProsmFile();

T tmp;

system("cls");

cout << "1)Добавить\n2)Найти\n3)Удалить\n4)Просмотреть\n5)изменить\n6)Записать в файл" << endl;

switch (\_getch())

{

case'1':

system("cls");

cin >> tmp;

obj.add(tmp);

break;

case'2':

system("cls");

obj.poisk(tmp);

break;

case'3':

system("cls");

obj.del();

break;

case'4':

system("cls");

obj.out();

break;

case'5':

system("cls");

obj.set();

break;

case'6':

system("cls");

obj.writefile();

break;

}

}

int main()

{

system("color F0");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

shablon<benzin> p\_b;

benzin obj\_b;

shablon<elecrobus>p\_e;

elecrobus obj\_e;

shablon<turist> p\_t;

turist obj\_t;

shablon<pasazhir> p\_p;

pasazhir obj\_p;

while (1)

{

cout << "======Вход======\n1)Пользователь\n2)Администратор" << endl;

rewind(stdin);

switch (\_getch())

{

case '1':

system("cls");

cout << "1)Бензиновые автобусы\n2)Електробусы\n3)Туристические автобусы\n4)Пасажирские автобусы\n0)выход" << endl;

rewind(stdin);

switch (\_getch())

{

case'1':

system("cls");

menupolz(p\_b);

break;

case'2':

system("cls");

menupolz(p\_e);

break;

case'3':

system("cls");

menupolz(p\_t);

break;

case'4':

system("cls");

menupolz(p\_p);

break;

case'0':

exit(0);

break;

}

system("pause");

system("cls");

case'2':

system("cls");

cout << "1)Бензиновые автобусы\n2)Електробусы\n3)Туристические автобусы\n4)Пасажирские автобусы\n0)выход" << endl;

rewind(stdin);

switch (\_getch())

{

case'1':

system("cls");

menuadmin(p\_b);

break;

case'2':

system("cls");

menuadmin(p\_e);

break;

case'3':

system("cls");

menuadmin(p\_t);

break;

case'4':

system("cls");

menuadmin(p\_p);

break;

case'0':

exit(0);

break;

}

system("pause");

system("cls");

}

}

return 0;

}

//pasazhir.cpp

#include"pasazhir.h"

#include "File.h"

pasazhir::pasazhir()

{

cena = 0;

}

int pasazhir::get\_cena()

{

return cena;

}

void pasazhir::settingsost()

{

cout << "изменить:\n1)Марку-\n2)Состояние-\n3)Количество мест-\n4)Гос. номер-\n5)Точка выезда-\n6)Точка приезда-\n7)Цена" << endl;

switch (\_getch())

{

case '1':

cout << "";

mark = letters();

break;

case '2':

cout << "";

sost = letters();

break;

case '3':

cout << "";

kmest = number();

break;

case '4':

cout << "Введите в формате 1111АА-1";

set\_gosnomer();

break;

case '5':

cout << "";

tochkaa = letters();

break;

case '6':

cout << "";

tochkab = letters();

break;

case '7':

cout << "";

cena = number();

break;

defaut:

cout << "" << endl;

break;

}

}

ostream& operator << (ostream& out, pasazhir& obj)

{

out << setiosflags(ios::left)

<< setw(15) << obj.get\_mark()

<< setw(15) << obj.get\_sost()

<< setw(15) << obj.get\_kmest()

<< setw(14) << obj.get\_gosnomer()

<< setw(20) << obj.get\_tochkaa()

<< setw(20) << obj.get\_tochkaa()

<< setw(20) << obj.get\_cena();

cout << endl;

return out;

};

istream& operator >> (istream& in, pasazhir& obj)

{

cout << "марка-";

obj.mark = letters();

cout << "Состояние-";

obj.sost = letters();

cout << "Кол-во мест-";

obj.kmest = number();

cout << "Гос. номер-" << endl;

cout << "Введите в формате 1111АА-1" << endl;

obj.set\_gosnomer();

cout << "Точка выезда-";

obj.tochkaa = letters();

cout << "Точка приезда-";

obj.tochkab = letters();

cout << "Цена проезда-";

obj.cena = number();

return in;

};

void pasazhir::shap()

{

cout << setiosflags(ios::left)

<< setw(15) << "Марка"

<< setw(15) << "Состояние"

<< setw(15) << "Кол-во мест"

<< setw(14) << "Гос. номер"

<< setw(20) << "Точка выезда"

<< setw(20) << "Точка приезда"

<< setw(20) << "Цена проезда";

cout << endl;

}

bool pasazhir::operator == (pasazhir& obj)

{

if ((obj.mark == this->mark || obj.mark == "n") &&

(obj.sost == this->sost || obj.sost == "n") &&

(obj.kmest == this->kmest || obj.kmest == 1) &&

(obj.gosnomer == this->gosnomer || obj.gosnomer == "n") &&

(obj.tochkaa == this->tochkaa || obj.tochkaa == "n") &&

(obj.tochkab == this->tochkab || obj.tochkab == "n") &&

(obj.cena == this->cena || obj.cena == 0))

return true;

else

return false;

}

ifstream& operator >> (ifstream& vvod, pasazhir& obj)

{

vvod >> obj.mark >> obj.sost >> obj.kmest >> obj.gosnomer >> obj.tochkaa >> obj.tochkab >> obj.cena;

return vvod;

}

ofstream& operator << (ofstream& vivod, pasazhir& obj)

{

vivod << obj.mark << ' ' << obj.sost << ' ' << obj.kmest << ' ' << obj.gosnomer << ' ' << obj.tochkaa << ' ' << obj.tochkab << ' ' << obj.cena << endl;;

return vivod;

}

//Shablon.cpp

#include "Shablon.h"

#include "File.h"

#include <fstream>

template<class T>

void shablon<T>::add(T p)

{

ochered<T>\* c = new ochered<T>;

c->object = p;

++size;

if (first == nullptr || last == nullptr)

{

c->right = nullptr;

c->left = nullptr;

first = c;

last = c;

}

else

{

c->left = first;

c->right = nullptr;

first->right = c;

first = c;

}

}

template<class T>

void shablon<T>::out()

{

T obj;

ochered<T>\* t;

obj.shap();

t = this->last;

int kk = 1;

while (t != nullptr)

{

cout << endl;

cout << kk << ")" << t->object << endl;

t = t->right;

++kk;

system("pause");

}

system("pause");

}

template<class T>

void shablon<T>::set()

{

T obj;

ochered<T>\* t;

obj.shap();

t = this->last;

while (t != nullptr)

{

t = this->last;

int kk = 1;

int p;

while (t != nullptr)

{

cout << kk << ")" << t->object << endl;

t = t->right;

++kk;

}

cout << "Номер изменяемого элемента" << endl<<endl;

cin >> p;

t = this->last;

for (int k = 1; k < p; k++)

{

t = t->right;

}

t->object.settingsost();

}

}

template<class T>

void shablon<T>::del()

{

T obj;

ochered<T>\* t;

obj.shap();

t = this->last;

int kk = 1;

int p;

while (t != nullptr)

{

cout << kk << ")" << t->object << endl;

t = t->right;

++kk;

}

cout << "номер удаляемого элемента" << endl;

cin >> p;

t = this->last;

for (int k = 1; k < p; k++)

{

t = t->right;

}

if (t == last)

last = t->right;

if (t == first)

first = t->left;

if (t->left != nullptr)

t->left->right = t->right;

if (t->right != nullptr)

t->right->left = t->left;

delete t;

t = last;

}

template<class T>

void shablon<T>::poisk(T obj)

{

ochered <T>\* p = last;

int i = 0;

cout << "Введите, что хотите найти" << endl;

cin >> obj;

while (i != size)

{

i++;

if (p->object == obj)

{

obj.shap();

cout << p->object << endl;

}

p = p->right;

}

system("pause");

}

template<class T>

void shablon<T>::writefile()

{

ofstream f(typeid(T).name());

if (!f.is\_open())

{

cout << "Файл не открыт" << endl;

}

else

this->addfile(f);

f.close();

}

template<class T>

void shablon<T>::addfile(ofstream& obj)

{

ochered<T>\* t = last;

int i = 0;

while (i!=size)

{

obj << t->object << endl;

t = t->right;

i++;

}

}

template <typename T>

void shablon<T> ::ProsmFile()

{

ifstream file2(typeid(T).name());

T obj = T();

while (file2 >> obj)

{

this->add(obj);

}

file2.close();

}

//turist.cpp

#include "turist.h"

#include "File.h"

turist::turist()

{

firma = "-";

}

string turist::get\_firma()

{

return firma;

}

void turist::settingsost()

{

cout << "изменить:\n1)Марку-\n2)Состояние-\n3)Количество мест-\n4)Гос. номер-\n5)Точка выезда-\n6)Точка приезда-\n7)Туристическая фирма" << endl;

switch (\_getch())

{

case '1':

cout << "";

mark = letters();

break;

case '2':

cout << "";

sost = letters();

break;

case '3':

cout << "";

kmest = number();

break;

case '4':

cout << "Введите в формате 1111АА-1";

set\_gosnomer();

break;

case '5':

cout << "";

tochkaa = letters();

break;

case '6':

cout << "";

tochkab = letters();

break;

case '7':

cout << "";

firma = letters();

break;

defaut:

cout << "" << endl;

break;

}

}

ostream& operator << (ostream& out, turist& obj)

{

out << setiosflags(ios::left)

<< setw(15) << obj.get\_mark()

<< setw(15) << obj.get\_sost()

<< setw(15) << obj.get\_kmest()

<< setw(14) << obj.get\_gosnomer()

<< setw(20) << obj.get\_tochkaa()

<< setw(20) << obj.get\_tochkab()

<< setw(20) << obj.get\_firma();

cout << endl;

return out;

};

istream& operator >> (istream& in, turist& obj)

{

cout << "Марка-";

obj.mark = letters();

cout << "Состояние-";

obj.sost = letters();

cout << "Кол-во мест-";

obj.kmest = number();

cout << "Гос. номер-" << endl;

cout << "Введите в формате 1111АА-1";

obj.set\_gosnomer();

cout << "Точка выезда-";

obj.tochkaa = letters();

cout << "Точка приезда-";

obj.tochkab = letters();

cout << "Тур. агенство-";

obj.firma = number();

return in;

};

void turist::shap()

{

cout << setiosflags(ios::left)

<< setw(15) << "Марка"

<< setw(15) << "Состояние"

<< setw(15) << "Кол-во мест"

<< setw(14) << "Гос. номер"

<< setw(20) << "Точка выезда"

<< setw(20) << "Точка приезда"

<< setw(20) << "Тур. агенство";

cout << endl;

}

bool turist::operator == (turist& obj)

{

if ((obj.mark == this->mark || obj.mark == "n") &&

(obj.sost == this->sost || obj.sost == "n") &&

(obj.kmest == this->kmest || obj.kmest == 1) &&

(obj.gosnomer == this->gosnomer || obj.gosnomer == "n") &&

(obj.tochkaa == this->tochkaa|| obj.tochkaa == "n") &&

(obj.tochkab == this->tochkab || obj.tochkab == "n") &&

(obj.firma == this->firma || obj.firma == "n"))

return true;

else

return false;

}

ifstream& operator >> (ifstream& vvod, turist& obj)

{

vvod >> obj.mark >> obj.sost >> obj.kmest >> obj.gosnomer >> obj.tochkaa >> obj.tochkab >> obj.firma;

return vvod;

}

ofstream& operator << (ofstream& vivod, turist& obj)

{

vivod << obj.mark << ' ' << obj.sost << ' ' << obj.kmest << ' ' << obj.gosnomer << ' ' << obj.tochkaa << ' ' << obj.tochkab << ' '<< obj.firma << endl;;

return vivod;

}

//Auto.h

#pragma once

#include "iostream"

#include "ostream"

#include "string"

#include "conio.h"

#include "iomanip"

#include "Windows.h"

using namespace std;

class Auto

{

protected:

string mark;

string sost;

int kmest;

string gosnomer;

public:

Auto()

{

mark = "-";

sost = "-";

kmest = 0;

gosnomer = "-";

}

string get\_mark()

{

return mark;

}

string get\_sost()

{

return sost;

}

int get\_kmest()

{

return kmest;

}

string get\_gosnomer()

{

return gosnomer;

}

string set\_gosnomer()

{

cin >> gosnomer;

return gosnomer;

}

virtual void shap()

{

}

};

//benzin.h

#pragma once

#include "gorodskoi.h"

#include "Exception.h"

class benzin : public gorodskoi

{

protected:

int rashodt;

public:

friend ofstream& operator << (ofstream& vivod, benzin& obj);

friend ifstream& operator >> (ifstream& vvod, benzin& obj);

friend ostream& operator << (ostream& out, benzin& obj);

friend istream& operator >> (istream& in, benzin& obj);

bool operator == (benzin& obj);

benzin();

int get\_rashodt();

void settingsost();

void shap();

};

//elecrobus.h

#pragma once

#include "gorodskoi.h"

#include "Exception.h"

class elecrobus : public gorodskoi

{

protected:

int rashode;

public:

friend ofstream& operator << (ofstream& vivod, elecrobus& obj);

friend ifstream& operator >> (ifstream& vvod, elecrobus& obj);

friend ostream& operator << (ostream& out, elecrobus& obj);

friend istream& operator >> (istream& in, elecrobus& obj);

bool operator == (elecrobus& obj);

elecrobus();

int get\_rashode();

void settingsost();

void shap();

};

//Exception.h

#pragma once

#include "Shablon.h"

using namespace std;

class Exception

{

private:

string strochka;

int numer;

public:

Exception(const Exception& q);

Exception(int \_numer, const string s);

~Exception();

void pokaz();

};

int number();

string letters();

//File.h

#pragma once

#include "fstream"

#include "iostream"

#include "Shablon.h"

using namespace std;

template <typename T>

class file

{

private:

ofstream file1;

ifstream file2;

public:

};

//gorodskoi.h

#pragma once

#include "Auto.h"

class gorodskoi : public Auto

{

protected:

string marshrut;

string kuzov;

public:

gorodskoi()

{

marshrut = "-";

kuzov= "-";

}

string get\_marshrut()

{

return marshrut;

}

string get\_kuzov()

{

return kuzov;

}

};

//mezgorod.h

#pragma once

#include "Auto.h"

class mezgorod : public Auto

{

protected:

string tochkaa;

string tochkab;

public:

mezgorod()

{

tochkaa = "-";

tochkab = "-";

}

string get\_tochkaa()

{

return tochkaa;

}

string get\_tochkab()

{

return tochkab;

}

};

//pasazhir.h

#pragma once

#include "mezgorod.h"

#include "Exception.h"

class pasazhir : public mezgorod

{

protected:

int cena;

public:

friend ofstream& operator << (ofstream& vivod, pasazhir& obj);

friend ifstream& operator >> (ifstream& vvod, pasazhir& obj);

friend ostream& operator << (ostream& out, pasazhir& obj);

friend istream& operator >> (istream& in, pasazhir& obj);

bool operator == (pasazhir& obj);

pasazhir();

int get\_cena();

void settingsost();

void shap();

};

//Shablon.h

#pragma once

#include "conio.h"

#include "File.h"

#include <iostream>

using namespace std;

template <class T>

struct ochered

{

T object;

ochered\* right;

ochered\* left;

};

template <class T>

class shablon

{

private:

int size;

ochered<T>\* first = nullptr;

ochered<T>\* last = nullptr;

public:

shablon()

{

first = nullptr;

last = nullptr;

size = 0;

}

~shablon()

{

}

void add(T p);

void poisk(T c);

void del();

void out();

void set();

void writefile();

void addfile(ofstream& f);

void ProsmFile();

};

//turist.h

#pragma once

#include "mezgorod.h"

#include "Exception.h"

class turist : public mezgorod

{

protected:

string firma;

public:

friend ofstream& operator << (ofstream& vivod, turist& obj);

friend ifstream& operator >> (ifstream& vvod, turist& obj);

friend ostream& operator << (ostream& out, turist& obj);

friend istream& operator >> (istream& in, turist& obj);

bool operator == (turist& obj);

turist();

string get\_firma();

void settingsost();

void shap();

};