#ifndef ANNUALREPORT\_H

#define ANNUALREPORT\_H

#include "TabOfIncome.h"

#include "TabOfExpenses.h"

class AnnualReport // Класс годового отчета

{

private: TabOfIncome\* ptrPL; //указатель на таблицу доходов

TabOfExpanses\* ptrER; //указатель на таблицу расходов

float expenses; //сумма расходоы

float income; //сумма доходов

public: AnnualReport(TabOfIncome\*, TabOfExpanses\*);

void display(); //отображение на экране

};

#endif

#ifndef AUTHENTICATION\_H

#define AUTHENTICATION\_H

#include "Profile.h"

class Authentication //экран ввода пользователя

{

private:

Profile\* ptrProfile; //указатель на пользователя

string addLogin; //логин

string addPassword; //пароль

int rules; //права доступа

public:

Authentication(Profile\* ptrUL) : ptrProfile(ptrUL)

{

}

int getRules(); //получить номер права доступа

void setUser(); //ввод пользователя

};

#endif

#ifndef CLIENT\_H

#define CLIENT\_H

#include <string>

using namespace std;

class Client // класс клиента

{

private:

string name; //ФИО

string reg\_num; //рег. номер машины владельца

string service; //услуга

string date; //дата обращения

public:

Client(string n, string rn, string s, string d);

~Client();

string getName(); //получить ФИО

string getRegNum();//получить рег. номер машины

string getService();//получить услугу

string getDate();//получить дату обащения

};

#endif

#ifndef DISPLAYSTABOFCLIENTS\_H

#define DISPLAYSTABOFCLIENTS\_H

#include <string>

#include "TabOfClients.h"

using namespace std;

class DisplayTabOfClients // экран ввода клиентов

{

private:

TabOfClients\* ptrTabOfClients; //указатель на список клиентов

string addName; //Фамилия И.О

string addRegNum; //Вид одежды

string addService; //Услуга

string addDate; //Дата обращения

public:

DisplayTabOfClients(TabOfClients\* ptrCL) : ptrTabOfClients(ptrCL)

{

}

void setClient(); //ввод клиента

};

#endif

#ifndef DISPLAYTABOFEXPENSES\_H

#define DISPLAYTABOFEXPENSES\_H

#include "TabOfExpenses.h"

class DisplayTabOfExpenses //класс экрана ввода расходов

{

private:

TabOfExpanses\* ptrExpenseList; //указатель на таблицу расходов

public: DisplayTabOfExpenses(TabOfExpanses\*);

void setExpense(); //ввести расходы

};

#endif

#ifndef DISPLAYTABOFINCOME\_H

#define DISPLAYTABOFINCOME\_H

#include "TabOfClients.h"

#include "TabOfIncome.h"

#include <string>

using namespace std;

class DisplayTabOfIncome //класс экрана ввода доходов

{

private:

TabOfIncome\* ptrTabOfIncome; //указатель на таблицу доходов

string addName; //ФИО клиента введное пользователем

float addPriceService; //стоимость услуги введная пользователем

float addPriceMaterials; //стоимость материалов введная пользователем

string addDatePrice; //дата оплаты введное пользователем

public:

DisplayTabOfIncome(TabOfIncome\* ptrPL) : ptrTabOfIncome(ptrPL)

{

}

void setIncome(); //ввод доходов

};

#endif

#ifndef EXPENSES\_H

#define EXPENSES\_H

#include <string>

using namespace std;

class Expenses //класс расходов

{

public:

string expenditure; //статья расхода

string recipient; //получатель платежа

float payment; //сумма платежа

string datePay; //дата оплаты

Expenses(){}

Expenses(string e, string r, float p, string d) : expenditure(e), recipient(r), payment(p), datePay(d)

{

}

};

#endif

#ifndef INCOME\_H

#define INCOME\_H

#include <string>

using namespace std;

class Income //Класс доходов

{

private:

string name; //имя клиента

float priceService; //стоимость услуги

float priceMaterials; //стоимость материалов

string datePrice;//дата оплаты

public:

Income(string, float, float, string);

string getName(); //получить имя

float getPriceService(); //получить стоимость услуги

float getPriceMaterials(); //получить стоимость материалов

string getDate(); //получить дату оплаты

};

#endif

#ifndef INTERFACE\_H

#define INTERFACE\_H

#include "TabOfClients.h"

#include "DisplayTabOfClients.h"

#include "Profile.h"

#include "Authentication.h"

#include "TabOfExpenses.h"

#include "DisplayTabOfExpenses.h"

#include "TabOfIncome.h"

#include "DisplayTabOfIncome.h"

#include "AnnualReport.h"

class Interface

{

private:

//указатели на таблицы и экраны ввода

Authentication\* ptrAuthentication;

Profile\* ptrUser;

DisplayTabOfClients\* ptrDisplayTabOfClients;

TabOfClients\* ptrTabOfClients;

TabOfIncome\* ptrTabOfIncome;

DisplayTabOfIncome\* ptrDisplayTabOfIncome;

TabOfExpanses\* ptrTabOfExpenses;

DisplayTabOfExpenses\* ptrDisplayTabOfExpenses;

AnnualReport\* ptrAnnualReport;

char ch;

public:

Interface();

~Interface();

void interact(); //основной интерфейс программы

};

#endif

#ifndef PROFILE\_H

#define PROFILE\_H

#include <string>

using namespace std;

class Profile // класс профиля пользователя

{

private:

string login; //логин

string password; //пароль

public:

Profile(string l, string p);

~Profile();

int access(string l, string p); //доступ

};

#endif

#ifndef TABOFCLIENTS\_H

#define TABOFCLIENTS\_H

#include <list>

#include "Client.h"

class TabOfClients //список клиентов

{

private:

list <Client\*> setPtrsClients; // контейнер список клиентов

list <Client\*>::iterator iter; //итератор

public: ~TabOfClients();

void insertClient(Client\*); //внесение клиента в таблицу

void display(); //отображение на экране

};

#endif

#ifndef TABOFEXPENSES\_H

#define TABOFEXPENSES\_H

#include <vector>

#include "Expenses.h"

class TabOfExpanses //класс таблицы расходов

{

private:

vector<Expenses\*> vectPtrsExpenses; //указатель контейнер вектор на расходы

vector<Expenses\*>::iterator iter; //итератор расходов

public:

~TabOfExpanses();

void insertExp(Expenses\*); //добавление в таблицу расходов

void display(); //отображение на экране

float getSumOfExpenses(); //расчитать сумму всех расходов

};

#endif

#ifndef TABOFINCOME\_H

#define TABOFINCOME\_H

#include <list>

#include "Income.h"

class TabOfIncome //таблица доходов

{

private:

list <Income\*> setPtrsRR; //указатель на доходы

list <Income\*>::iterator iter; //итератор

public:

~TabOfIncome();

void insertIncome(Income\*); //добавление в таблицу доходов

void display(); //отображение на экране

float getSumOfIncomes(); //сумма всех доходов

};

#endif