**ПРИЛОЖЕНИЕ №1**

**13) Написание заголовочных файлов**

AnnualReport.h

#ifndef ANNUALREPORT\_H

#define ANNUALREPORT\_H

#include "ProfitList.h"

#include "ExpenseList.h"

class AnnualReport // Класс годового отчета

{

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

ExpenseList\* ptrER; //указатель на таблицу расходов

float expenses; //сумма расходоы

float prof; //сумма доходов

public: AnnualReport(ProfitList\*, ExpenseList\*);

void display(); //отображение на экране

};

#endif

Client.h

#ifndef CLIENT\_H

#define CLIENT\_H

#include <string>

using namespace std;

class Client // класс клиента

{

private:

string name; //фамилия и.о

string clother; //вид одежды

string service; //услуга

string date; //дата обращения

public:

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

~Client();

string getName(); //получить имя

string getClother();//получить вид одежды

string getService();//получить услугу

string getDate();//получить дату обащения

};

#endif

ClientInputScreen.h

#ifndef CLIENTINPUTSCREEN\_H

#define CLIENTINPUTSCREEN\_H

#include <string>

#include "ClientList.h"

using namespace std;

class ClientInputScreen // экран ввода клиентов

{

private:

ClientList\* ptrClientList; //указатель на список клиентов

string addName; //Фамилия И.О

string addClother; //Вид одежды

string addService; //Услуга

string addDate; //Дата обращения

public:

ClientInputScreen(ClientList\* ptrCL) : ptrClientList(ptrCL)

{

}

void setClient(); //ввод клиента

};

#endif

ClientList.h

#ifndef CLIENTLIST\_H

#define CLIENTLIST\_H

#include <list>

#include "Client.h"

class ClientList //список клиентов

{

private:

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

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

public: ~ClientList();

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

void display(); //отображение на экране

};

#endif

Expense.h

#ifndef EXPENSE\_H

#define EXPENSE\_H

#include <string>

using namespace std;

class Expense //класс расходов

{

public:

string category; //категория услуги

string payee; //название организации

string datePay; //дата оплаты

float amount; //сумма оплаты

Expense(){}

Expense(string c, string p, string d, float a) : category(c), payee(p), datePay(d), amount(a)

{

}

};

#endif

ExpenseInputScreen.h

#ifndef EXPENSEINPUTSCREEN\_H

#define EXPENSEINPUTSCREEN\_H

#include "ExpenseList.h"

class ExpenseInputScreen //класс экрана ввода расходов

{

private:

ExpenseList\* ptrExpenseList; //указатель на таблицу расходов

public: ExpenseInputScreen(ExpenseList\*);

void setExpense(); //ввести расходы

};

#endif

ExpenseList.h

#ifndef EXPENSELIST\_H

#define EXPENSELIST\_H

#include <vector>

#include "Expense.h"

class ExpenseList //класс таблицы расходов

{

private:

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

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

public:

~ExpenseList();

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

void display(); //отображение на экране

float getSumOfExpenses(); //расчитать сумму всех расходов

};

#endif

Gets.h

#ifndef GETS\_H

#define GETS\_H

#include <string>

using namespace std;

void getaLine(string& inStr); //функция для ввода строки с ограничением ввода

char getaChar(); //функция для ввода символов с ограничением ввода

#endif

Profit.h

#ifndef PROFIT\_H

#define PROFIT\_H

#include <string>

using namespace std;

class Profit //Класс доходов

{

private:

string name; //имя клиента

float priceCloth; //стоимость стирки

float priceService; //стоимость услуги

string datePrice;//дата оплаты

public:

Profit(string, float, float, string);

float getPriceCloth(); //получить стоимость стирки

string getName(); //получить имя

float getPriceService(); //получить стоимость услуги

string getDate(); //получить дату оплаты

};

#endif

ProfitInputScreen.h

#ifndef PROFITINPUTSCREEN\_H

#define PROFITINPUTSCREEN\_H

#include "ClientList.h"

#include "ProfitList.h"

#include <string>

using namespace std;

class ProfitInputScreen //класс экрана ввода доходов

{

private:

ProfitList\* ptrProfitList; //указатель на таблицу доходов

string addName; //имя клиента введное пользователем

float addPriceCloth; //стоимость стирки введная пользователем

float addPriceService; //стоимость услуги введная пользователем

string addDatePrice; //дата оплаты введное пользователем

public:

ProfitInputScreen(ProfitList\* ptrPL) : ptrProfitList(ptrPL)

{

}

void setProfit(); //ввод доходов

};

#endif

ProfitList.h

#ifndef PROFITLIST\_H

#define PROFITLIST\_H

#include <list>

#include "Profit.h"

class ProfitList //таблица доходов

{

private:

list <Profit\*> setPtrsRR; //список указатель на доходы

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

public:

~ProfitList();

void insertProfit(Profit\*); //добавление в таблицу доходов

void display(); //отображение на экране

float getSumOfProfits(); //сумма всех доходов

};

#endif

stdafx.h

#pragma once

#include "targetver.h"

#include <iomanip>

#include <stdio.h>

#include <tchar.h>

#include <iostream>

#include <vector>

#include <list>

#include <string>

#include <numeric>

#include <clocale>

#include "Gets.h"

using namespace std;

User.h

#ifndef USER\_H

#define USER\_H

#include <string>

using namespace std;

class User // класс пользователя

{

private:

string login; //логин

string password; //пароль

public:

User(string l, string p);

~User();

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

};

#endif

UserInputScreen.h

#ifndef USERINPUTSCREEN\_H

#define USERINPUTSCREEN\_H

#include "User.h"

class UserInputScreen //экран ввода пользователя

{

private:

User\* ptrUser; //указатель на пользователя

string addLogin; //логин

string addPassword; //пароль

int rules; //права доступа

public:

UserInputScreen(User\* ptrUL) : ptrUser(ptrUL)

{

}

int getRules(); //получить номер права доступа

void setUser(); //ввод пользователя

};

#endif USERINPUTSCREEN\_H

UserInterface.h

#ifndef USERINTERFACE\_H

#define USERINTERFACE\_H

#include "UserInputScreen.h"

#include "AnnualReport.h"

#include "ExpenseInputScreen.h"

#include "ExpenseList.h"

#include "ProfitInputScreen.h"

#include "ProfitList.h"

#include "ClientInputScreen.h"

#include "ClientList.h"

#include "UserInputScreen.h"

#include "User.h"

class UserInterface

{

private:

//указатели на таблицы и экраны ввода

UserInputScreen\* ptrUserInputScreen;

User\* ptrUser;

ClientList\* ptrClientList;

ClientInputScreen\* ptrClientInputScreen;

ProfitList\* ptrProfitList;

ProfitInputScreen\* ptrProfitInputScreen;

ExpenseList\* ptrExpenseList;

ExpenseInputScreen\* ptrExpenseInputScreen;

AnnualReport\* ptrAnnualReport;

char ch;

public:

UserInterface();

~UserInterface();

void interact(); //основной интерфейс программы

};

#endif