**Заголовочный файл AnnualReport.h**

#ifndef ANNUALREPORT\_H

#define ANNUALREPORT\_H

#include "IncomeList.h"

#include "ExpenseList.h"

class AnnualReport // Класс годового отчета

{

private: IncomeList\* ptrInc; //записи заработка

ExpenseList\* ptrER; //записи раcходов

float expenses; //сумма расходов

float inc; //сумма заработка

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

void display(); //вывод годового отчёта

};

#endif

**Заголовочный файл Client.h**

#ifndef CLIENT\_H

#define CLIENT\_H

#include <string>

using namespace std;

class Client // класс клиента

{

private:

string name; //фамилия и.о

string flower; //цветок

string service; //вид услуги

string date; //дата обращения

public:

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

~Client();

string getName(); //получить имя

string getFlower();//получить цветок

string getService();//получить вид услуги

string getDate();//получить дату обащения

};

#endif

**Заголовочный файл 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 инициалы

string addFlower; //Цветок

string addService; //Услуга

string addDate; //Дата обращения

public:

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

{

}

void setClient(); //ввод клиента

};

#endif

**Заголовочный файл 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

**Заголовочный файл Expense.h**

#ifndef EXPENSE\_H

#define EXPENSE\_H

#include <string>

using namespace std;

class Expense //класс расходов

{

public:

string category; //категория услуги

string payee; //кому платим (арендадатели, коммунальщики)

string dateOfPayment; //когда произведена оплата

float amount; //сколько оплачено

Expense(){}

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

{

}

};

#endif

**Заголовочный файл ExpenseInputScreen.h**

#ifndef EXPENSEINPUTSCREEN\_H

#define EXPENSEINPUTSCREEN\_H

#include "ExpenseList.h"

class ExpenseInputScreen //класс экрана ввода расходов

{

private:

ExpenseList\* ptrExpenseList; //указатель на список расходов

public:

ExpenseInputScreen(ExpenseList\*);

void inputExpense(); //ввод расходов

};

#endif

**Заголовочный файл ExpenseList.h**

#ifndef EXPENSELIST\_H

#define EXPENSELIST\_H

#include <vector>

#include "Expense.h"

class ExpenseList //класс списка расходов

{

private:

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

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

public:

~ExpenseList();

void insertExp(Expense\*); //занесение расходов в список

void display(); //вывод на экран

float getAllExpenses(); // подсчёт всех расходов

};

#endif

**Заголовочный файл Getf.h**

#ifndef GETF\_H

#define GETF\_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

**Заголовочный файл Income.h**

#ifndef INCOME\_H

#define INCOME\_H

#include <string>

using namespace std;

class Income //Класс заработка

{

private:

string flower; //название цветка

float costOfFlower; //цена цветка

float costOfService; //цена услуги

string dateOfPayment;//дата оплаты

public:

Income(string, float, float, string);

string getFlower(); //получить название цветка

float getCostOfFlower(); //получить цену цветка

float getCostOfService(); //получить цену услуги

string getDateOfPayment(); //получить дату оплаты

};

#endif

**Заголовочный файл IncomeInputScreen.h**

#ifndef INCOMEINPUTSCREEN\_H

#define INCOMEINPUTSCREEN\_H

#include "ClientList.h"

#include "IncomeList.h"

#include <string>

using namespace std;

class IncomeInputScreen //класс экрана ввода доходов

{

private:

IncomeList\* ptrIncomeList; //указатель на список доходов

string inputFlower; //ввод названия цветка

float inputCostOfFlower; //ввод цена цветка

float inputCostOfService; //ввод цены услуги

string inputDateOfPayment; //ввод даты платежа

public:

IncomeInputScreen(IncomeList\* ptrInc) : ptrIncomeList(ptrInc)

{

}

void inputIncome(); //ввод доходов

};

#endif

**Заголовочный файл IncomeList.h**

#ifndef INCOMELIST\_H

#define INCOMELIST\_H

#include <list>

#include "Income.h"

class IncomeList //Список доходов

{

private:

list <Income\*> setPtrsIncome; //список указатель на заработок

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

public:

~IncomeList();

void insertIncome(Income\*); //добавление в список заработка

void display(); //отображение на экране

float getSumOfIncome(); //подсчёт всего заработка

};

#endif

**Заголовочный файл stdafx.h**

// stdafx.h : include file for standard system include files,

// or project specific include files that are used frequently, but

// are changed infrequently

#pragma once

#include "targetver.h"

#include <iomanip>

#include <stdio.h>

#include <tchar.h>

#include <iostream>

#include <vector>

#include <list>

#include <string>

#include <numeric>

#include <locale>

#include <clocale>

#include "Getf.h"

using namespace std;

// TODO: reference additional headers your program requires here

**Заголовочный файл UserInterface.h**

#ifndef USERINTERFACE\_H

#define USERINTERFACE\_H

#include "AnnualReport.h"

#include "ExpenseInputScreen.h"

#include "ExpenseList.h"

#include "IncomeInputScreen.h"

#include "IncomeList.h"

#include "ClientInputScreen.h"

#include "ClientList.h"

class UserInterface

{

private:

ClientList\* ptrClientList; //указатель на клиента

ClientInputScreen\* ptrClientInputScreen; //указатель на ввод клиента

IncomeList\* ptrIncomeList; //указатель на заработок

IncomeInputScreen\* ptrIncomeInputScreen; //указатель на ввод заработка

ExpenseList\* ptrExpenseList; //указатель на расходы

ExpenseInputScreen\* ptrExpenseInputScreen; //указатель на ввод расхода

AnnualReport\* ptrAnnualReport; //указатель на годовой расчёт

char ch;

public:

UserInterface();

~UserInterface();

void interact(); //главная рабочая область

};

#endif