Приложение А

**AnnualReport.h**

///////////////////////// КЛАСС ГОДОВОГО ОТЧЕТА ///////////////////

#ifndef ANNUALREPORT\_H

#define ANNUALREPORT\_H

#include "IncomeRecord.h"

#include "ExpenseRecord.h"

class AnnualReport

{

private: IncomeRecord\* ptrIN; //указатель на таблицу доходов

ExpenseRecord\* ptrER; //указатель на таблицу расходов

float expenses, prof; //сумма расходоы и доходов

public: AnnualReport(IncomeRecord\*, ExpenseRecord\*);

void display(); //отображение на экране

};

#endif

**Expense.h**

///////////// КЛАСС РАСХОДОВ /////////////////////

#ifndef EXPENSE\_H

#define EXPENSE\_H

#include <string>

using namespace std;

class Expense //класс расходов

{

public:

string category; //категория услуги или товара

string org; //название организации

string datePay; //дата оплаты

float sum; //сумма оплаты

Expense(){}

Expense(string c, string o, string d, float s) : category(c), org(o), datePay(d), sum(s) // конструктор с параметрами

{

}

};

#endif

**ExpenseRecord.h**

///// КЛАСС ТАБЛИЦЫ РАСХОДОВ ////////

#ifndef EXPENSERECORD\_H

#define EXPENSERECORD\_H

#include <vector>

#include "Expense.h"

#include <clocale>

class ExpenseRecord

{

private:

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

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

public:

~ExpenseRecord();

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

void display(); //отображение на экране

float displaySummary(); // для годового отчета

};

#endif

**ExpenseInputScreen.h**

////// КЛАСС ЭКРАНА ВВОДА РАСХОДОВ ///////

#ifndef EXPENSEINPUTSCREEN\_H

#define EXPENSEINPUTSCREEN\_H

#include "ExpenseRecord.h"

class ExpenseInputScreen

{

private:

ExpenseRecord\* ptrExpenseList; //указатель на таблицу расходов

public: ExpenseInputScreen(ExpenseRecord\*);

void setExpense(); //ввести расходы

};

#endif

**Income.h**

////////// КЛАСС ДОХОДОВ ////////

#ifndef INCOME\_H

#define INCOME\_H

#include <string>

using namespace std;

class Income

{

private:

string id; //id продавца

float priceTov; //стоимость товаров

float priceService; //стоимость услуги

string datePrice;//дата оплаты

public:

Income(string, float, float, string);

float getPriceTov(); //получить стоимость товара

string getID(); //получить ID

float getPriceService();

string getDate();

};

#endif

**IncomeRecord.h**

///// КЛАСС ТАБЛИЦЫ ДОХОДОВ ////////

#ifndef INCOMERECORD\_H

#define INCOMERECORD\_H

#include <list>

#include "Income.h"

class IncomeRecord

{

private:

list <Income\*> setPtrsIN; //список указатель на доходы

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

public:

~IncomeRecord();

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

void display(); //отображение на экране

float getSumOfIncome(); //сумма всех доходов для годового отчета

};

#endif

**IncomeInputScreen.h**

//////// КЛАСС ЭКРАНА ВВОДА ДОХОДОВ ///////

#ifndef INCOMEINPUTSCREEN\_H

#define INCOMEINPUTSCREEN\_H

#include "ProdavciList.h"

#include "IncomeRecord.h"

#include <string>

using namespace std;

class IncomeInputScreen

{

private:

IncomeRecord\* ptrIncomeR; //указатель на таблицу доходов

string ID; //имя клиента введное пользователем

float PriceTov; //стоимость стирки введная пользователем

float PriceService; //стоимость услуги введная пользователем

string DatePrice; //дата оплаты введное пользователем

public:

IncomeInputScreen(IncomeRecord\* ptrIR) : ptrIncomeR(ptrIR)

{

}

void setIncome(); //ввод доходов

};

#endif

**Interface.h**

////////////////////// ГЛАВНЫЙ КЛАСС ДЛЯ ЗАПУСКА /////////////

#ifndef INTERFACE\_H

#define INTERFACE\_H

#include "Prodavci.h" // класс инициализации

#include "TheFirstScreen.h" // вывод на экран "Введите логин и пароль"

#include "Prodaza.h"

#include "ProdavciList.h"

#include "AnnualReport.h"

#include "ExpenseInputScreen.h"

#include "ExpenseRecord.h"

#include "IncomeInputScreen.h"

#include "IncomeRecord.h"

class Interface {

private: // раздел указтелей на таблицы и экраны

char pr, h;

TheFirstScreen\* ptrTheFirstScreen;

Prodavci\* ptrProdavci;

ProdavciList\* ptrProdavciList;

Prodaza\* ptrProdaza;

IncomeRecord\* ptrIncomeRecord;

IncomeInputScreen\* ptrIncomeInputScreen;

ExpenseRecord\* ptrExpenseRecord;

ExpenseInputScreen\* ptrExpenseInputScreen;

AnnualReport\* ptrAnnualReport;

public: // методы

Interface(); // конструктор

~Interface(); // деструктор

void menu(); //основное меню программы

};

#endif

**Prodavci.h**

////////////////////// КЛАСС ИНИЦИАЛИЗАЦИИ ПОЛЬЗОВАТЕЛЯ /////////////////

#ifndef PRODAVCI\_H

#define PRODAVCI\_H

#include <string>

using namespace std;

class Prodavci

{

private:

char username; //идентификатор пользователя

string password; //пароль

public:

Prodavci(char u, string p); // конструктор с параметрами

~Prodavci(); // деструктор

int avtorizaciya(char u, string p); // получение доступа

};

#endif

**ProdavciList.h**

/// КЛАСС СПИСКА СОТРУДНИКОВ НА СМЕНЕ /////////////

#ifndef PRODAVCILIST\_H

#define PRODAVCILIST\_H

#include <list>

#include "Sotrydniki.h"

class ProdavciList //список id

{

private:

list <Sotrydniki\*> setPtrsSotr; // контейнер список сотрудников

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

public: ~ProdavciList();

void insertSotr(Sotrydniki\*); //внесение сотрудников в список

void display(); //отображение на экране

};

#endif

**Prodaza.h**

//////// КЛАСС УЧЕТА СОТРУДНИКОВ НА СМЕНЕ /////

#ifndef PRODAZA\_H

#define PRODAZA\_H

#include <string>

#include "ProdavciList.h"

using namespace std;

class Prodaza

{

private:

ProdavciList\* ptrProdavciList; //указатель на список сотрудников

string Number; // id - продавца

string fio;

string dol; // Вид должности

string zp;

public:

Prodaza(ProdavciList\* ptrPL) : ptrProdavciList(ptrPL)

{

}

void prodazaProd();

};

#endif

**Sotrydniki.h**

//// КЛАСС СОТРУДНИКОВ /////////

#ifndef SOTRYDNIKI\_H

#define SOTRYDNIKI\_H

#include <string>

using namespace std;

class Sotrydniki

{

private:

string number; // id - продавца

string fio; //фио продавца

string katprod; //категория продавца

string date; //дата смены

public:

Sotrydniki(string n, string f, string k, string d);

~Sotrydniki();

string getNumber(); //получение id

string getFio();//получение фио продавца

string getKat();//получение категории продавца

string getDate();//получение зарплаты

};

#endif

**TheFirstScreen.h**

//////////////////////// ЭКРАН АВТОРИЗАЦИИ ////////////////

#pragma once

#ifndef THEFIRSTSCREEN\_H

#define THEFIRSTSCREEN\_H

#include "Prodavci.h"

class TheFirstScreen

{

private:

Prodavci\* ptrname; //указатель на использующего

char username1; //идентификатор

string password1; // пароль

int prava; // право доступа

public:

TheFirstScreen(Prodavci\* ptrPC) : ptrname(ptrPC){} // конструктор

int getprava(); // получить номер права доступа

void inicial(); //ввод пользователя

};

#endif