

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Новосибирский государственный университет экономики и управления «НИНХ»**

**(ФГБОУ ВО «НГУЭУ», НГУЭУ)**

Кафедра информационных технологий

Консольное приложение «Заказы на производство»

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

Дисциплина: Языки программирования

Ф.И.О студента: Нищаков Максим Вячеславович

Направление: 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль): Программная инженерия

Номер группы: ФИ202

Номер зачетной книжки: 220132

Номер варианта расчетно-графической работы: 2

Проверил: Ковригин Алексей Викторович – кандидат педагогических наук, доцент

Новосибирск 2024

# **Содержание**

[**Содержание 2**](#_Toc155562774)

[**Выполнение и реализация поставленной задачи 3**](#_Toc155562775)

[**Глава 1. Контроллер меню заказов 4**](#_Toc155562776)

[**Глава 2.1 Fabrika 4**](#_Toc155562777)

[**Глава 2.2 Класс client 7**](#_Toc155562778)

[**Глава 2.3 Класс record 8**](#_Toc155562779)

[**Глава 2.4 Основная программа 11**](#_Toc155562780)

# 

# **Выполнение и реализация поставленной задачи**

Поставленная задача – написание консольного приложения на языке программирования С++.

Программа должна уметь выполнять следующие задачи:

1. Создавать 3 объекта сущности.
2. Указывать взаимодействия объектов сущности.
3. Иметь интерфейс ввода и вывода из программы.
4. Возможность редактирования сущностей.
5. Ручной ввод сущностей.
6. Сохранение и выгрузка сущностей из структурированного текстового файла.
7. Удаление выбранных сущностей.

# **Глава 1. Контроллер меню заказов**

Этот код – это контроллер меню заказов. Он создает объект класса controller. Метод execute() выполняет цикл, пока пользователь не введет 8, что означает выход из программы. В каждой итерации цикла пользователю предлагается выбрать команду из меню. В зависимости от выбранной команды выполняется соответствующий метод объекта current\\_service.

Класс controller не содержит никаких данных, только методы. Метод menu() выводит меню заказов. Метод execute() выполняет цикл до тех пор, пока пользователь не введет 8. В каждой итерации цикла пользователю предлагается выбрать команду из меню. В зависимости от выбранной команды выполняется соответствующий метод объекта current\\_service.

Класс current\\_service представляет собой фабрику, которая создает объекты для работы с заказами. Метод print() выводит все заказы. Метод addRc() добавляет новый заказ. Метод deleteRc(n) удаляет заказ с номером n. Метод saveToFile(b) сохраняет заказы в файл, где b - это булева переменная, указывающая, нужно ли сохранять только текущие заказы или все заказы. Метод loadFile() загружает заказы из файла. Метод changeRecord(n) изменяет заказ с номером n.

Метод menu() выводит меню заказов. Он выводит список команд, которые пользователь может выбрать. Ввод пользователя проверяется в цикле, пока он не введет корректный номер. Когда пользователь вводит номер, выполняется соответствующий метод объекта current\\_service.

# **Глава 2.1 Fabrika**

Этот код представляет собой фабрику для создания объектов заказа. Класс fabrika создает объект current\\_service. Метод print() выводит все заказы. Метод addRc() добавляет новый заказ.

В конструкторе класса fabrika создается массив объектов заказа с заданными параметрами. Эти параметры включают тип товара, имя, фамилию, номер телефона и компанию покупателя. Затем создается объект current\\_service, который будет использоваться для работы с заказами.

Метод print() выводит все заказы, проходя по массиву объектов заказа. Метод addRc() запрашивает у пользователя тип товара, количество партий и количество штук в партии. Затем он создает новый объект заказа с этими параметрами и добавляет его в массив объектов заказа.

Класс current\\_service не содержит никаких данных, только методы. Метод print() выводит все заказы. Метод addRc() добавляет новый заказ.

cout << "Укажите ваше имя: ";

cin >> name;

cout << "Укажите вашу фамилию: ";

cin >> surname;

cout << "Укажите ваш номер телефона для дальнейшей связи: ";

cin >> number;

cout << "Укажите вашу компанию. Если нет, поставьте '-': ";

cin >> company;

record newrc(name, surname, number,company, object, amount, numb);

record\* newor = new record[n + 1];

for (int i = 0; i < n; i++) {

newor[i] = orders[i];

}

newor[n] = newrc;

n++;

orders = newor;

}

void addRcFile(record &newrc) {

record\* newor = new record[n + 1];

for (int i = 0; i < n; i++) {

newor[i] = orders[i];

}

newor[n] = newrc;

n++;

orders = newor;

}

void deleteRc(int k) {

record\* newor = new record[n - 1];

for (int i = 0,j=0; i < n; i++,j++) {

if (i == (k - 1))i++;

if (i<n)newor[j] = orders[i];

}

n--;

orders = newor;

}

void saveToFile(bool app) {

ofstream out;

try {

if (app)

out.open("orders.txt", ios::app);

else out.open("orders.txt");

}

catch (const char\* error\_message) {

cout << error\_message << endl;

return;

}

if (out.is\_open())

{

string rcInfo;

for (int i = 0; i < n; i++) {

record cr\_rc = orders[i];

rcInfo = cr\_rc.getClient().getName() + "" + cr\_rc.getClient().getSurname() + "" + cr\_rc.getClient().getNumber() + "" + cr\_rc.getClient().getCompany() + "" + cr\_rc.getObject() + "" + to\_string(cr\_rc.getAmount()) + "" + to\_string(cr\_rc.getNumber());

out << rcInfo << endl;

}

out.close();

}

}

void loadFile() {

ifstream in("orders.txt");

if (in.is\_open())

{

string newrc;

while (getline(in, newrc)) {

// вывод текста из файла

string name = newrc.substr(0, newrc.find(""));

newrc = newrc.substr(newrc.find("") + 2, newrc.length() - newrc.find(""));

string surname = newrc.substr(0, newrc.find(""));

newrc = newrc.substr(newrc.find("") + 2, newrc.length() - newrc.find(""));

string number = newrc.substr(0, newrc.find(""));

newrc = newrc.substr(newrc.find("") + 2, newrc.length() - newrc.find(""));

string company = newrc.substr(0, newrc.find(""));

newrc = newrc.substr(newrc.find("") + 2, newrc.length() - newrc.find(""));

string object = newrc.substr(0, newrc.find(""));

newrc = newrc.substr(newrc.find("") + 2, newrc.length() - newrc.find(""));

int amount = stoi(newrc.substr(0, newrc.find("")));

newrc = newrc.substr(newrc.find("") + 2, newrc.length() - newrc.find(""));

int numb = stoi(newrc.substr(0, newrc.length()));

record\* newrc = new record(name,surname, number,company, object, amount, numb);

addRcFile(\*newrc);

}

}

in.close();

}

void changeRecord(int n) {

orders[n - 1].change();

}

int getN() {

return n;

}

};

# **Глава 2.2 Класс client**

Этот код представляет класс client, который представляет собой объект клиента. Класс имеет приватные переменные для имени, фамилии, номера телефона и названия компании. Есть конструктор по умолчанию, который инициализирует переменные значениями по умолчанию, и конструктор, который принимает параметры для инициализации переменных.

Метод print() выводит объект клиента в консоль. Объект клиента может быть создан с помощью конструктора или через создание нового объекта client с заданными параметрами. class client {

private:

string name, surname, number,company;

public:

string getName() {

return name;

}

string getSurname() {

return surname;

}

string getNumber() {

return number;

}

string getCompany() {

return company;

}

client() {

name = "Ваня";

surname = "Иванов";

number = "88005553535";

company = "OOO";

}

client(string name, string surname, string number,string company) {

this->name = name;

this->surname = surname;

this->number = number;

this->company = company;

}

void print() {

cout << surname << " " << name << " - " << number <<" "<<company<< endl;;

}

};

Переменные экземпляра класса client защищены, что означает, что они доступны только для методов этого класса. Это обеспечивает безопасность и контроль над данными клиента.

# **Глава 2.3 Класс record**

# Этот код представляет класс record, который представляет собой объект заказа. Класс имеет приватные переменные для объекта заказа, количества партий, количества штук в партии и объекта клиента. Есть конструктор по умолчанию и конструктор, который принимает параметры для инициализации переменных.

# Метод print() выводит объект заказа в консоль, включая информацию о клиенте. Метод change() позволяет пользователю изменять различные аспекты заказа, такие как объект заказа, количество партий, количество штук в партии и информацию о клиенте.

# Класс record обеспечивает простой способ создания и управления объектами заказа, а также предоставляет удобный интерфейс для взаимодействия с ними.

#include <iostream>

#include <string>

#include <stdlib.h>

#include "people.h"

#include <time.h>

using namespace std;

class record {

private:

string type\_object[4] = {

"Кольца",

"Браслеты",

"Сережки",

"Цепочки"

};

string object;

int number;

int amount;

client cl;

public:

record() {

object = "Кольца";

amount = 10;

number = 100;

}

record(string name, string surname, string number, string company, string object,int amount,int numb) {

this->cl = \*(new client(name, surname, number,company));

this->object = object;

this->amount = amount;

this->number = numb;

}

string getObject() {

return object;

}

int getAmount() {

return amount;

}

int getNumber() {

return number;

}

client getClient() {

return cl;

}

string getClientInfo() {

cl.print();

}

void print() {

cl.print();

cout << "Товар: " << object << "; Количество партий: " << amount << ". Количество в одной партии: " <<number<<endl;

}

void change() {

cout << "Выберите пункт заказа, который вы хотели бы отредактировать:\n";

cout << "1 - Товар\n";

cout << "2 - Количество партий\n";

cout << "3 - Количество штук товара в одной парти\n";

cout << "4 - Информацию о клиенте\n";

int choose = 0;

while (choose < 1 choose>4){

cin >> choose;

}

switch (choose){

case 1:

{

cout << "Выберите тип товара:" << endl;

int select = 0;

for (int i = 0; i < 4; i++) {

cout << "[" << i + 1 << "] " << type\_object[i] << endl;

}

while (select < 1 select>4) {

cin >> select;

}

object = type\_object[select - 1];

break;}

case 2:

{

cout << "Укажите количество партий товара: ";

cin >> amount;

break;

}

case 3:

{

int select = 0;

cout << "Укажите количество штук товара в одной партии:" << endl;

do

{

cin >> number;

} while (number < 1);

break;

}

case 4:

{

string name, surname, number, company;

cout << "Укажите имя нового клиента: ";

cin >> name;

cout << "Укажите фамилию нового клиента: ";

cin >> name;

cout << "Укажите номер телефона нового клиента для дальнейшей связи: ";

cin >> number;

cout << "Укажите компанию заказчика, если она есть (- если нет)";

cin >> company;

cl = client(name, surname, number,company);

break;

}

default:

break;

}

}

};

# **Глава 2.4 Основная программа**

Этот код включает заголовочный файл controller.h, который содержит объявление класса controller. Затем он включает заголовочный файл record.h, который содержит объявление класса record.

Функция main() начинается с инициализации генератора случайных чисел и установки кодировки консоли на кириллицу. Затем создается объект controller, и вызывается его метод execute().

Класс controller представляет собой класс, который управляет системой заказов. Он содержит методы для отображения меню заказов, обработки выбора пользователя и вызова соответствующих методов объекта current\\_service.

Метод execute() — это цикл, который продолжается до тех пор, пока пользователь не введет 8. В каждой итерации цикла пользователю предлагается выбрать команду из меню. В зависимости от выбранной команды выполняется соответствующий метод объекта current\\_service.

Класс current\\_service представляет собой фабрику, которая создает объекты для работы с заказами. Метод print() выводит все заказы. Метод addRc() добавляет новый заказ. Метод deleteRc(n) удаляет заказ с номером n. Метод saveToFile(b) сохраняет заказы в файл, где b — это булева переменная, указывающая, нужно ли сохранять только текущие заказы или все заказы. Метод loadFile() загружает заказы из файла. Метод changeRecord(n) изменяет заказ с номером n.

Метод menu() выводит меню заказов. Он выводит список команд, которые пользователь может выбрать. Ввод пользователя проверяется в цикле, пока он не введет корректный номер. Когда пользователь вводит номер, выполняется соответствующий метод объекта current\\_service.

#include <iostream>

#include "controller.h"

#include <stdlib.h>

#include <time.h>

#include <windows.h>

using namespace std;

int main() {

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

controller ctrl;

ctrl.execute();

}