



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения
(ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: «Объектно-ориентированное программирование»
по профилю: Информатизация организаций
направления профессиональной подготовки: Прикладная информатика,
бакалавриат

Тема: Разработка программы для ведения учета заказов клиентов ресторана

Студент : Подчуфаров Глеб Андреевич

Группа: ИНБО-04-18

Работа представлена к защите _____ (дата) _____ / _____ /
(подпись и ф.и.о.
студента)

Руководитель: Хлебникова Валерия Леонидовна

Работа допущена к защите _____ (дата) _____ / _____ /
(подпись ф.и.о.
руководителя)

Оценка по итогам защиты: _____

_____/ _____ /
_____/ _____ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей,
принявших защиту)

М. МИРЭА. 2019г.

УДК 004.432.4

Подчуфаров Г.А. Разработка программы для построения графиков функций/ Курсовая работа по дисциплине «Объектно-ориентированное программирование» профиля «Информатизация организаций» направления профессиональной подготовки бакалавра 09.03.03. «Прикладная информатика» (2ой семестр) / руководитель асс. Хлебникова В.Л./ кафедра ИППО Института ИТ МИРЭА – с. 37, илл. 17, ист. 6, (в т.ч. 2 на англ. яз.).

Целью работы является разработка и реализация программы для ведения учета заказов клиентов ресторана.

The aim of the work is to develop and implement a program for keeping records of restaurant customers' orders.

Ключевые слова: курсовая работа, бакалавриат, программирование, программа, ресторан, заказ, блюдо, позиция, отчет.

М. МИРЭА. Ин-т ИТ. Каф. ИППО. 2018г. @ Подчуфаров Г.А

ВВЕДЕНИЕ	4
ТЕХНИЧЕСКОЕ ЗАДАНИЕ	4
2.1 Общие положения	5
2.2 Назначение и цели создания системы	Ошибка! Закладка не определена.
2.3 Характеристика объекта автоматизации	Ошибка! Закладка не определена.
2.4 Требования к системе	Ошибка! Закладка не определена.
2.5 Состав и содержание работ по созданию (развитию) системы	Ошибка! Закладка не определена.
2.6 Порядок контроля и приемки системы	Ошибка! Закладка не определена.
2.7 Требование к составу и содержанию работ по подготовке объекта к вводу системы в действие	Ошибка! Закладка не определена.
2.8 Требования к документированию	14
ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ	15
3.9 Проектирование программы	15
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	Ошибка! Закладка не определена.
ЗАКЛЮЧЕНИЕ	21
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	23
ПРИЛОЖЕНИЕ №1. ИСХОДНЫЙ КОД ПРОГРАММЫ	24
ПРИЛОЖЕНИЕ №2. UML-ДИАГРАММА	Ошибка! Закладка не определена.

ВВЕДЕНИЕ

Сегодня мы живем в мире, в котором компьютерные технологии развиваются огромными шагами.

Такое развитие привело к необходимости создания компьютерных систем, которые облегчали бы людям однообразную работу с различными документами. На практике, каждая такая система реализуется с помощью специализированного программного обеспечения.

Такие системы используются в сферах, связанных с ресторанным бизнесом. К примеру, простейшая программа может быть полезна в кафе, ресторанах для упрощения процесса ведения отчетности. Именно такой программе посвящена данная курсовая работа.

Ни для кого не секрет, что главная цель отчетов в местах общественного питания – контроль за продукцией и заказами. Поэтому крайне важно использовать самые эффективные методы ведения отчетности, включая связанные с информационными технологиями.

В ходе выполнения курсовой работы должно быть проведено освоение или углубление общекультурных и профессиональных компетенций, предусмотренных федеральным образовательным стандартом, а именно: ОПК-1, ПК-14, ПК-20.

Выполнение курсовой работы должно опираться на положения СМКО МИРЭА 7.5.1/04.И.05-16 «Инструкция по организации и проведению курсового проектирования» от 06.12.2016.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

2.1 Общие положения

2.1.1 Полное наименование системы и ее условное обозначение

Полное наименование: Разработка программы для учёта заказов клиентов ресторана.

2.1.2 Номер договора (контракта)

Работа выполняется на основании Задания на выполнение курсовой работы.

2.1.3 Наименования организации-заказчика и организаций-участников работ

Заказчик: РТУ МИРЭА

Исполнитель: Подчуфаров Глеб Андреевич студент группы ИНБО-04-

18

2.1.4 Перечень документов, на основании которых создается система

Документы, на основании которых создается система:

– Учебный план (№1382.9 09.03.03 ПИ Очн, пс, 4 года (8 сессий) УМУ_09.03.03_ИО_ИИТ_2018).

2.1.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работы по созданию системы: 26 февраля 2019 года.

Плановый срок окончания работы по созданию системы: 20 мая 2019 года.

2.1.6 Источники и порядок финансирования работ

Разработка Системы финансируется РТУ МИРЭА, а, в частности,

Федеральным бюджетом Российской Федерации.

2.1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы

Система передаётся в виде готового программного обеспечения на базе средств вычислительной техники Заказчика в сроки, установленные в п. 2.1.5 данного ТЗ. Приёмка осуществляется в составе Исполнителя и уполномоченных представителей Заказчика.

Порядок предъявления системы, её испытаний и окончательной приёмки определён в п.7 настоящего ТЗ. Совместно с предъявлением Системы Исполнителем производится сдача разработанного комплекта документации согласно п. 9 данного ТЗ.

2.1.8 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

При разработке системы Исполнитель должен руководствоваться требованиями следующих документов:

- ГОСТ 34.601-90 Комплекс стандартов Автоматизированные системы.

Стадии создания;

- методические указания по выполнению курсовой работы для бакалавров.

2.2 Назначение и цели создания системы

2.2.1 Назначение системы

Система предназначена для упрощения процесса ведения отчетности в местах общественного питания.

2.2.2 Цели создания системы

Цели создания Системы:

- демонстрация прикладного программного обеспечения, работающего с графической библиотекой;
- сдача курсовой работы;
- приобретение опыта разработки приложений средней сложности;
- изучение объектно-ориентированного программирования;
- приобретение навыков написания технической документации.

2.3 Характеристика объекта автоматизации

Объектом автоматизации данной системы является ПО для упрощения процесса ведения отчетности в местах общественного питания.

2.4 Требования к системе

2.4.1 Требования к системе в целом

2.4.1.1 Требования к структуре и функционированию системы

2.4.1.1.1 Перечень подсистем, их назначение и основные характеристики

Разрабатываемая система, должна включать класс для работы с графическим окном и файлами: MainWindow;

2.4.1.1.2 Требования к способам и средствам связи для информационного обмена между компонентами системы

Связь между компонентами системы обеспечивается функциями, определёнными в компонентах системы.

2.4.1.2 Требования к численности и квалификации персонала системы

Для работы разрабатываемой системы необходимо достаточно одного человека.

К работе с системой должны допускаться сотрудники, имеющие навыки работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

2.4.1.3 Показатели назначения

Система должна моделировать ПО для ведения ресторанного бизнеса. Система должна позволять создавать отчеты по наличию продукции на складе, а так же по заказам за определенный промежуток времени. Требования ко времени выполнения данных функций не предъявляются.

2.4.1.4 Требования к надежности

Отсутствуют.

2.4.1.5 Требования к безопасности

Отсутствуют.

2.4.1.6 Требования к эргономике и технической эстетике

Графический интерфейс Системы должен разрабатываться на основе офтальмологических исследований, с использованием сочетаний цветов, комфортных для глаз человека.

2.4.1.7 Требования к транспортабельности для подвижных АС

Разрабатываемая Система должна запускаться на любых устройствах под управлением операционной системы не младше чем Windows 10 без процедуры установки или повторной компиляции данной Системы.

2.4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Для корректной работы разрабатываемая Система должна храниться в отдельном каталоге со всеми своими компонентами. Особых требований по обслуживанию данная Система не имеет.

2.4.1.9 Требования к защите информации от несанкционированного доступа

Требования к какой-либо защите информации в данной системе не предъявляются.

2.4.1.10 Требования по сохранности информации при авариях

Отсутствуют.

2.4.1.11 Требования к защите от влияния внешних воздействий

Защита от влияния внешних воздействий должна обеспечиваться средствами программно-технического комплекса Заказчика.

2.4.1.12 Требования к патентной чистоте

Данная Система должна являться интеллектуальной собственностью Исполнителя и быть патентно чистой по отношению ко всем странам мира.

2.4.1.13 Требования по стандартизации и унификации

Система должна быть реализована методами объектно-ориентированного программирования. Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое

отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

2.4.1.14 Дополнительные требования

Дополнительные требования не предъявляются.

2.4.2 Требования к функциям (задачам), выполняемым системой

Система должна выполнять следующие функции:

- отображение списков продуктов, блюд и заказов на экране;
- автоматический расчет себестоимости блюд, счета заказов;
- автоматическое формирование отчетов по кол-ву продуктов на складе, а также по заказам за промежуток времени или по одному заказу;
- автоматическое формирование меню из доступных блюд;

2.4.3 Требования к видам обеспечения

2.4.3.1 Требования к математическому обеспечению системы

Отсутствуют.

2.4.3.2 Требования информационному обеспечению системы

Отсутствуют.

2.4.3.3 Требования к лингвистическому обеспечению системы

Для создания Системы должен быть использован высокоуровневый язык программирования C++. Для взаимодействия Системы с пользователем,

руководства по использованию Системы и документации Системы должен использоваться русский язык.

2.4.3.4 Требования к программному обеспечению системы

Данная Система должна работать на операционных системах не младше Windows 10.

2.4.3.5 Требования к техническому обеспечению

Используемое при разработке программное обеспечение и библиотеки программных кодов должны иметь широкое распространение, быть общедоступными и использоваться в промышленных масштабах. Базовой программной платформой должна являться операционная система Windows.

2.4.3.6 Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

2.4.3.7 Требования к организационному обеспечению

Организационное обеспечение системы должно быть достаточным для эффективного выполнения персоналом возложенных на него обязанностей при осуществлении автоматизированных и связанных с ними неавтоматизированных функций системы.

К работе с системой должны допускаться сотрудники, имеющие навыки работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

2.4.3.8 Требования к методическому обеспечению

Данная система должна поставляться с определённым пакетом документации, состоящем из:

- технического задания (ГОСТ 34.602.89);

- пояснительной записки (ГОСТ 19.201-78);
- задания на выполнение курсовой работы;
- руководства пользователя.

2.5 Состав и содержание работ по созданию (развитию) систем

Таблица 1 Содержание Этапов работ

№ этапа	Содержание работ	Срок
1	Разработка структуры входных и выходных данных	29.02.19-10.03.19
2	Разработка алгоритма решения задачи	11.03.19-20.03.19
3	Разработка структуры программы	08.03.19-30.03.19
4	Разработка тестов	03.04.19-07.04.19
5	Написание текста программы	15.04.19-28.04.19
6	Отладка программы	25.04.19-06.05.19
7	Написание и оформление отчета о выполненной в виде работе пояснительной записки	06.05.19-17.05.19

2.6 Порядок контроля и приемки системы

2.6.1 Виды, состав, объем и методы испытаний системы

Для разработанной системы будут сформированы тесты для проверки функций записи позиций и формирования отчетов. Тестовые входные записи

будут иметь корректные выходные данные, сравнение выходных данных системы, с которыми покажет правильность реализации алгоритмов.

2.6.2 Общие требования к приемке работ по стадиям

Разработка данной Системы делится на шесть стадий:

- получение задания на выполнение курсовой работы;
- составление и согласование технического задания;
- создание и тестирование Системы Исполнителем;
- написание технической документации для Системы;
- демонстрация Системы Заказчику;
- защита курсовой работы.

2.6.3 Статус приемочной комиссии

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

2.7 Требование к составу и содержанию работ по подготовке объекта к вводу системы в действие

В ходе выполнения проекта на объекте автоматизации требуется выполнить работы по подготовке к вводу системы в действие. При подготовке к вводу в эксплуатацию системы должен обеспечить выполнение следующих работ:

- Определить подразделение и ответственных должностных лиц, ответственных за внедрение и проведение опытной эксплуатации системы;

- Обеспечить присутствие пользователей на обучении работе с системой, проводимом Исполнителем;
- Обеспечить соответствие помещений и рабочих мест пользователей системы в соответствии с требованиями, изложенными в настоящем ЧТЗ;
- Обеспечить выполнение требований, предъявляемых к программно-техническим средствам, на которых должно быть развернуто программное обеспечение;
- Совместно с Исполнителем подготовить план развертывания системы на технических средствах Заказчика;
- Привести поступающую в систему информацию (в соответствии с требованиями к информационному и лингвистическому обеспечению) к виду, пригодному для обработки с помощью ЭВМ;
- Провести опытную эксплуатацию системы.

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие, включая перечень основных мероприятий и их исполнителей должны быть уточнены на стадии подготовки рабочей документации и по результатам опытной эксплуатации.

2.8 Требования к документированию

Кроме создания работоспособной Системы Исполнитель должен составить пакет документации, состоящий из:

- технического задания;
- пояснительной записки.

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ

3.9 Проектирование программы

3.9.1 Функциональные требования к программе

Функциональные требования к программе описаны в пункте 2.4.2 Технического задания.

3.9.2 Проектирование классов

Программа написана на языке программирования C++, с использованием библиотеки SFML. Для реализации функциональной части системы был реализован класс (Рис. 1):

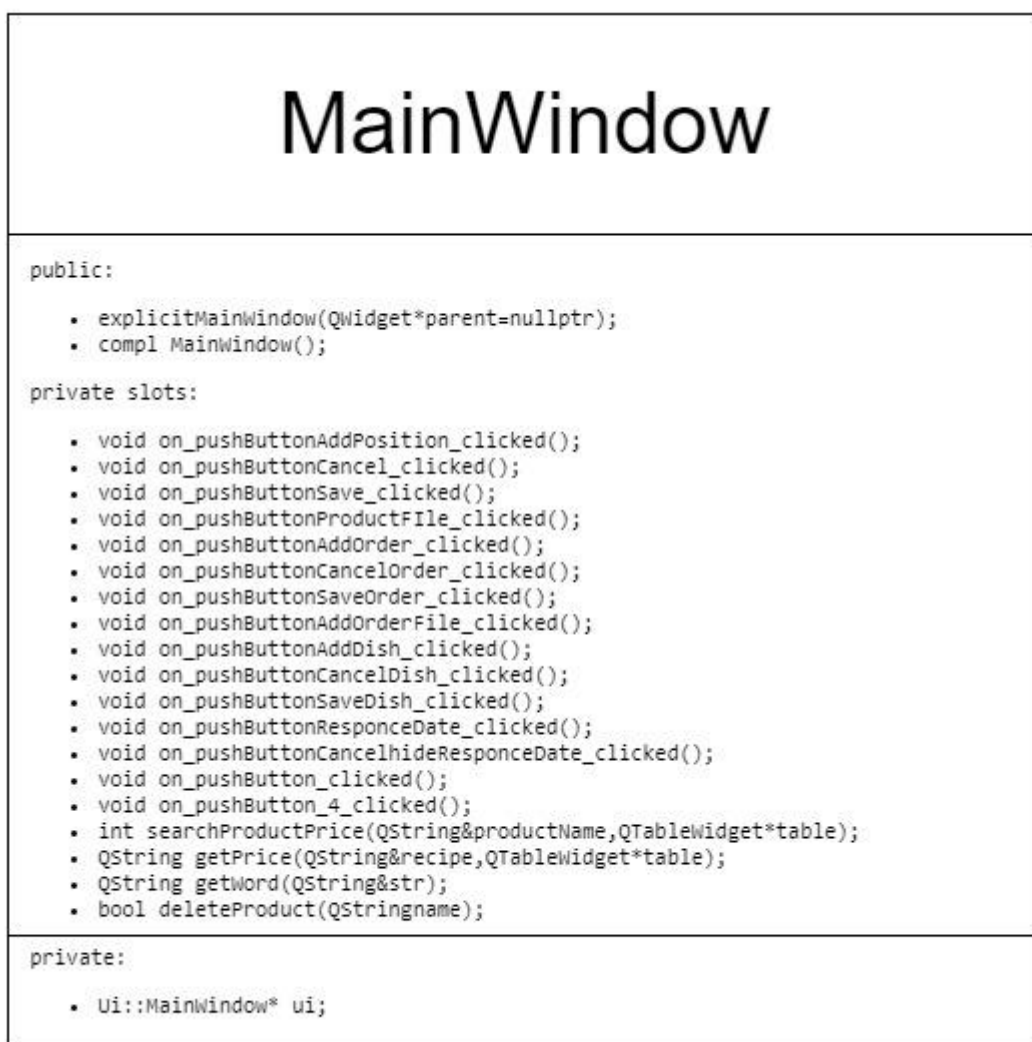


Рис. 1 Общая UML диаграмма

Этот класс Main Window отвечает, как за взаимодействие пользователем с графической системой, так и за обработку таблиц с позициями и за формирование отчетов.

3.9.3 Разработка программы

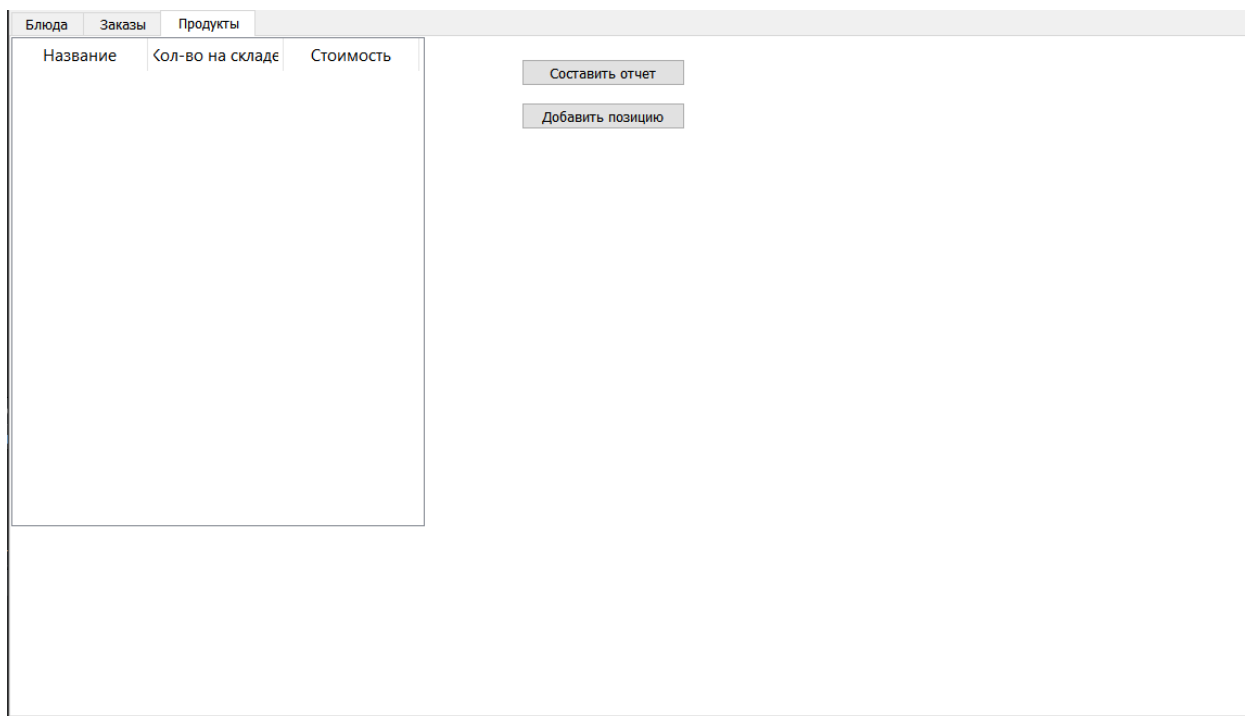
3.9.3.1 Описание проектного решения

При запуске программы сначала необходимо заполнить каталог доступных продуктов, затем каталог блюд. При этом себестоимость этих блюд будет рассчитана автоматически.

После пользователь может приступать к добавлению заказов. Программа автоматически рассчитает счет заказа.

После добавления заказа его можно сохранить в файл или сохранить все заказы за какой-либо промежуток времени.

3.9.3.2 Запуск и работа программы



На рисунке демонстрируется результат запуска программы

MainWindow

Блюда Заказы **Продукты**

Название	Кол-во на складе	Стоимость
----------	------------------	-----------

Составить отчет

Название продукта

Кол-во на складе

Себестоимость

Отмена Сохранить

Добавление позиции

MainWindow

Блюда Заказы **Продукты**

	Название	Кол-во на складе	Стоимость
1	meat	12	100

Составить отчет

Добавить позицию

Демонстрация добавленной позиции

MainWindow

Блюда Заказы Продукты

	Название	Рецепт блюда	Цена	Себестоимость
1	example	meat	500	100

Составить меню

Отмена Сохранить

Автоматический расчёт себестоимости блюда.

MainWindow

Блюда Заказы Продукты

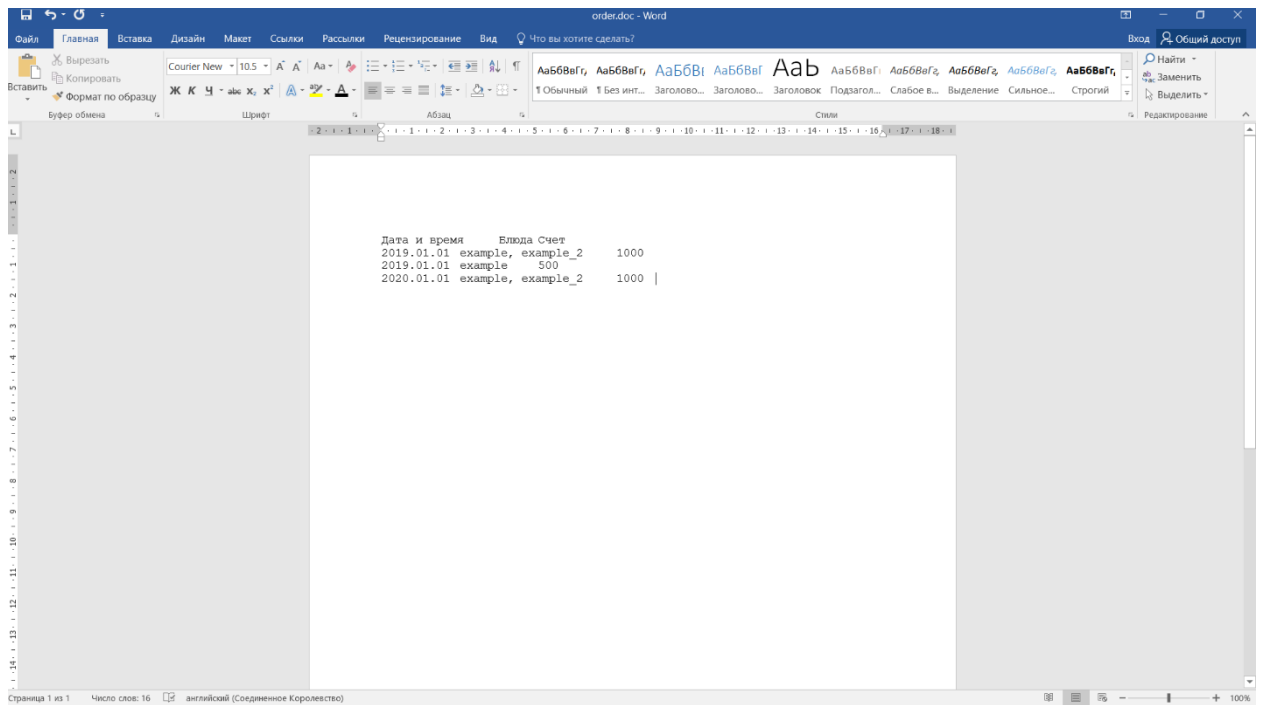
	Дата	Блюда в заказе	Счет заказа
1	2019.01.01	example, example_2	1000

Заказы за период

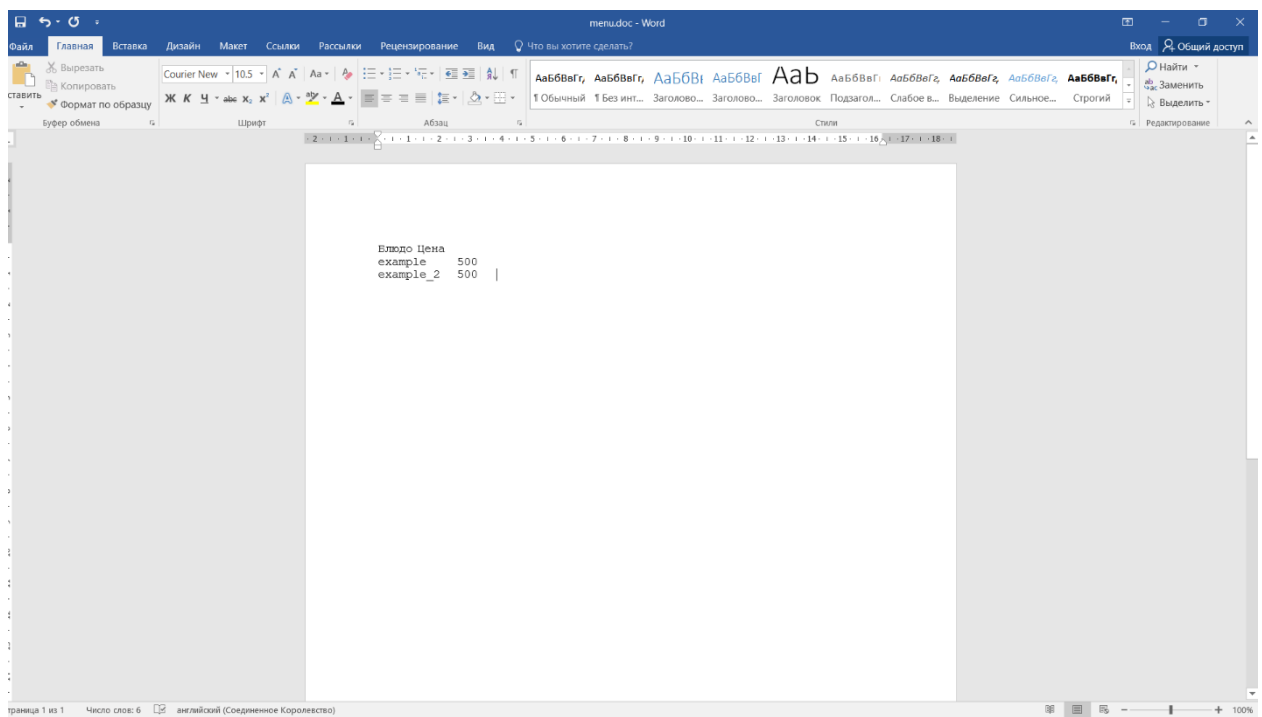
Добавить заказ

Сохранить заказ в отчет

Расчет счета заказа



Отчет, составленный программой



Меню составленное программой

3.9.3.2 Тестирование программы

Программа была протестирована путем проверки всех ее функциональных возможностей. Системные функции работают корректно. Ошибок системы не выявлено. Все выходные данные соответствуют ожиданиям.

ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ ПО

1. Необходимо заполнить каталог продуктов и блюд.
2. После можно приступать к работе с заказами.
3. При необходимости составления меню, необходимо нажать соответствующую кнопку в вкладке «Блюда».
4. Для составления отчетов по заказам во вкладке «Заказы» необходимо нажать соответствующую кнопку, в случае выбора составления отчетов за промежуток времени введите даты этого промежутка в соответствующие поля.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы был разработан программный продукт, удовлетворяющий досуг пользователей. При разработке программного продукта были построены UML диаграммы, на основе, которых реализованы классы в самой программе. Для реализации методов классов были использованы структуры данных из STL. Для проверки корректности результатов, получаемых после исполнения алгоритмов, были сформированы выходные тестовые данные. Код проекта разбит на отдельные файлы на основе логики работы отдельных частей кода. Для удобства работы с функционалом картотеки клиентов был реализован пользовательский графический интерфейс. Все требования, предъявленные в Техническом задании, удовлетворены.

Разработка приложения позволила систематизировать все знания, накопленные за пройденный курс, и обобщила уже имеющиеся. Разработка программного средства помогла разобраться в основных тонкостях программирования на языке C++, а также понять принципы объектно-ориентированного стиля.

В ходе выполнения курсовой работы было проведено освоение и углубление общекультурных и профессиональных компетенций, предусмотренных федеральным образовательным стандартом, а именно: ОПК-1, ПК-14, ПК-20.

При выполнении курсовой работы и оформлении отчета по ней были использованы положения СМКО МИРЭА 7.5.1/04.И.05-16 «Инструкция по организации и проведению курсового проектирования» от 06.12.2016.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

В качестве источников разработки использовались данные ресурсы:

1) Зорина, Н.В. Методические указания по выполнению курсовой работы для бакалавров, обучающихся по направлениям 09.03.02 «Информационные системы и технологии», 09.03.04 «Программная инженерия» / Н.В. Зорина, Л.Б. Зорин, О.В. Соболев,- Москва, 2017 - 41 с.

2) Лафоре Р. Объектно-ориентированное программирование C++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2018. – 928 с.: ил. – (Серия «Классика Computer Science»).

3) <https://www.sfml-dev.org/documentation/2.5.1/>

4) Мейерс, Скотт Эффективный и современный C++. 42 рекомендации по использованию C++11 и C++14 / Скотт Мейерс. - М.: Вильямс, 2015. - 304 с.

5) Страуструп, Бьерн Дизайн и эволюция C++ / Бьерн Страуструп. - М.: ДМК Пресс, 2016. - 446 с.

Приложение №1. Исходный код программы

Файл MainWindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QTableWidget>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_pushButtonAddPosition_clicked();
    void on_pushButtonCancel_clicked();
    void on_pushButtonSave_clicked();
    void on_pushButtonProductFile_clicked();

    void on_pushButtonAddOrder_clicked();
    void on_pushButtonCancelOrder_clicked();
    void on_pushButtonSaveOrder_clicked();
    void on_pushButtonAddOrderFile_clicked();

    void on_pushButtonAddDish_clicked();
    void on_pushButtonCancelDish_clicked();
    void on_pushButtonSaveDish_clicked();
    void on_pushButtonResponseDate_clicked();
    void on_pushButtonCancelhideResponseDate_clicked();

    void on_pushButton_clicked();
    void on_pushButton_4_clicked();

    int searchProductPrice(QString& productName, QTableWidget* table);
    QString getPrice(QString& recipe, QTableWidget* table);
    QString getWord(QString& str);
    bool deleteProduct(QString name);

private:
    Ui::MainWindow *ui;
```

```
};
```

```
#endif // MAINWINDOW_H
```

Исходники

Файл MainWindow.cpp

```
%:include "mainwindow.h"
%:include "ui_mainwindow.h"
%:include <QMessageBox>
#include <QDebug>
#include <QFile>
#include <QDate>

#define print qDebug() <<

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    on_pushButtonCancel_clicked();
    on_pushButtonCancelOrder_clicked();
    on_pushButtonCancelDish_clicked();
    on_pushButtonCancelhideResponceDate_clicked();

}

MainWindow::~MainWindow()
{
    for(auto i =0; i<ui->tableProduct->columnCount();i++)
        for(auto j=0;j<ui->tableProduct->rowCount();j++)
            delete ui->tableProduct->item(i,j);

    for(auto i =0; i<ui->tableOrder->columnCount();i++)
        for(auto j=0;j<ui->tableOrder->rowCount();j++)
            delete ui->tableOrder->item(i,j);

    for(auto i =0; i<ui->tableDish->columnCount();i++)
        for(auto j=0;j<ui->tableDish->rowCount();j++)
            delete ui->tableDish->item(i,j);

    delete ui;
}

void MainWindow::on_pushButtonSave_clicked()
{
    ui->tableProduct->setRowCount(ui->tableProduct->rowCount()+1);

    ui->tableProduct->setItem(ui->tableProduct->rowCount()-1,
        0,
        new QTableWidgetItem(ui->lineEditName-
>text()));
    ui->tableProduct->setItem(ui->tableProduct->rowCount()-1,
        1,
        new QTableWidgetItem(ui->lineEditCount-
>text()));
    ui->tableProduct->setItem(ui->tableProduct->rowCount()-1,
        2,
        new QTableWidgetItem(ui->lineEditPrice-
>text()));
}
```

```

        on_pushButtonCancel_clicked();
    }

void MainWindow::on_pushButtonSaveDish_clicked()
{
    ui->tableDish->setRowCount(ui->tableDish->rowCount()+1);

    ui->tableDish->setItem(ui->tableDish->rowCount()-1,
                        0,
                        new QTableWidgetItem(ui->lineEditNumDish-
>text()));
    ui->tableDish->setItem(ui->tableDish->rowCount()-1,
                        1,
                        new QTableWidgetItem(ui->lineEditRecipeDish-
>text()));
    QString recipe = ui->lineEditRecipeDish->text();

    ui->tableDish->setItem(ui->tableDish->rowCount()-1,
                        3,
                        new QTableWidgetItem(getPrice(recipe, ui-
>tableProduct)));
    ui->tableDish->setItem(ui->tableDish->rowCount()-1,
                        2,
                        new QTableWidgetItem(ui->lineEditPriceDish-
>text()));
    on_pushButtonCancelDish_clicked();
}

void MainWindow::on_pushButtonSaveOrder_clicked()
{
    ui->tableOrder->setRowCount(ui->tableOrder->rowCount()+1);

    ui->tableOrder->setItem(ui->tableOrder->rowCount()-1,
                        0,
                        new QTableWidgetItem(ui->lineEditDateOrder-
>text()));
    ui->tableOrder->setItem(ui->tableOrder->rowCount()-1,
                        1,
                        new QTableWidgetItem(ui->lineEditDishOrder-
>text()));
    QString dish = ui->lineEditDishOrder->text();

    ui->tableOrder->setItem(ui->tableOrder->rowCount()-1,
                        2,
                        new QTableWidgetItem(getPrice(dish, ui-
>tableDish)));
    on_pushButtonCancelOrder_clicked();
}

bool MainWindow::deleteProduct(QString nameDish)
{
    bool error = false;

    QString recipe;
    for(auto i =0; i<ui->tableDish->rowCount();i++){
        if(ui->tableDish->item(i,0)->text().remove(' ') == nameDish.remove('
')) {
            recipe = ui->tableDish->item(i,1)->text().remove(' ');
            break;
        }
    }

    if(!recipe.size()) return true;
}

```

```

        while(recipe.size()){
            QString productName = getWord(recipe);
            for(auto i =0; i<ui->tableProduct->rowCount();i++){
                if(ui->tableProduct->item(i,0)->text().remove(' ') ==
productName) {
                    int count = ui->tableProduct->item(i,1)->text().toInt() - 1;
                    if(count < 0) {
                        count = 0;
                        error = !count;
                    }

                    ui->tableProduct->item(i,1)->setText(QString::number(count));
                }
            }
            if(error) return error;
        }
        return error;
    }
}

QString MainWindow::getPrice(QString& recipe, QTableWidgetItem* tableForSearch)
{
    recipe.remove(' ');
    int dishPrice = 0;
    bool error = false;
    while(recipe.size() && !error){
        QString nameDishOrProduct = getWord(recipe);
        int price = searchProductPrice(nameDishOrProduct, tableForSearch);
        if(price != -1) dishPrice += price;
        else error = true;

        if(tableForSearch==ui->tableDish && !error)
        {
            error = deleteProduct(nameDishOrProduct);
        }
    }

    if(error) return "Ошибка, проверьте введенные данные";
    else return QString::number(dishPrice);
}

int MainWindow::searchProductPrice(QString& productName, QTableWidgetItem* table)
{
    for(auto i =0; i<table->rowCount();i++){
        if(table->item(i,0)->text().remove(' ') == productName) {
            return table->item(i,2)->text().toInt();
        }
    }
    return -1;
}

QString MainWindow::getWord(QString& str)
<%
    QString word;

    int i=0;

    while(i < str.size() && str[i] != ',')
    <%
        word += str[i];
        i++;
    }

    if(i<str.size())

```

```

        str.remove(0,i+1);
    else
        str.remove(0,i);
    return word;
}

void MainWindow::on_pushButtonAddDish_clicked()
{
    ui->pushButtonSaveDish->show();
    ui->pushButtonCancelDish->show();

    ui->lineEditNumDish->show();
    ui->lineEditPriceDish->show();
    ui->lineEditRecipeDish->show();

    ui->pushButtonAddDish->hide();
}

void MainWindow::on_pushButtonCancelDish_clicked()
{
    ui->pushButtonSaveDish->hide();
    ui->pushButtonCancelDish->hide();

    ui->lineEditNumDish->hide();
    ui->lineEditPriceDish->hide();
    ui->lineEditRecipeDish->hide();

    ui->pushButtonAddDish->show();
}

void MainWindow::on_pushButtonAddPosition_clicked()
{
    ui->pushButtonSave->show();
    ui->pushButtonCancel->show();
    ui->lineEditName->show();
    ui->lineEditPrice->show();
    ui->lineEditCount->show();

    ui->pushButtonAddPosition->hide();
}

void MainWindow::on_pushButtonCancel_clicked()
{
    ui->pushButtonSave->hide();
    ui->pushButtonCancel->hide();
    ui->lineEditName->hide();
    ui->lineEditPrice->hide();
    ui->lineEditCount->hide();

    ui->pushButtonAddPosition->show();
}

void MainWindow::on_pushButtonAddOrder_clicked()
{
    ui->pushButtonSaveOrder->show();
    ui->pushButtonCancelOrder->show();

    ui->lineEditDIshOrder->show();
    ui->lineEditDateOrder->show();

    ui->pushButtonAddOrder->hide();
}

```

```

void MainWindow::on_pushButtonCancelOrder_clicked()
{
    ui->pushButtonSaveOrder->hide();
    ui->pushButtonCancelOrder->hide();

    ui->lineEditDishOrder->hide();
    ui->lineEditDateOrder->hide();

    ui->pushButtonAddOrder->show();
}

void MainWindow::on_pushButtonProductFile_clicked()
{
    QFile file("tableProduct.doc");

    if (!file.open(QIODevice::WriteOnly | QIODevice::Text))
    {
        print "Ошибка"; return;
    }

    QTextStream out(&file);

    out << QString("Название продукта\tКол-во на
склад\tСтоимость\n").toUtf8();
    for(auto i =0; i<ui->tableProduct->rowCount();i++){
        for(auto j=0;j < ui->tableProduct->columnCount();j++){
            out << QString(ui->tableProduct->item(i,j)->text() +
'\t').toUtf8();
            out << QString("\n").toUtf8();
        }

        file.close();
    }

void MainWindow::on_pushButton_clicked()
{
    QFile file("menu.doc");

    if (!file.open(QIODevice::WriteOnly | QIODevice::Text))
    {
        print "Ошибка"; return;
    }

    QTextStream out(&file);

    out << QString("Блюдо\tЦена\n").toUtf8();
    for(auto i =0; i<ui->tableDish->rowCount();i++){
        for(auto j=0;j < ui->tableDish->columnCount();j+=2){
            out << QString(ui->tableDish->item(i,j)->text() + '\t').toUtf8();
            out << QString("\n").toUtf8();
        }

        file.close();
    }

QDate getDate(QString str)
{
    return QDate(str.toInt()/10000,
                (str.toInt()/100)%100,
                str.toInt() % 100);
}

void MainWindow::on_pushButton_4_clicked()

```



```

{
    ui->pushButtonResponseDate->show();
    ui->lineEdit_3->show();
    ui->lineEdit_4->show();
    ui->pushButtonCancelhideResponseDate->show();

    ui->pushButton_4->hide();
}

void MainWindow::on_pushButtonResponseDate_clicked()
{
    QPair <QDate, QDate> date(getDate(ui->lineEdit_3->text().remove('.')),
                              getDate(ui->lineEdit_4->text().remove('.')));

    QFile file("order.doc");

    if (!file.open(QIODevice::WriteOnly | QIODevice::Text))
    {
        print "Ошибка"; return;
    }

    QTextStream out(&file);
    out << QString("Дата и время\tБлюда\tСчет\n").toUtf8();

    for(auto i =0; i<ui->tableOrder->rowCount();i++){
        if(getDate(ui->tableOrder->item(i,0)-
>text().remove('.')).toJulianDay() >= date.first.toJulianDay() &&
        getDate(ui->tableOrder->item(i,0)-
>text().remove('.')).toJulianDay() <= date.second.toJulianDay())
        {
            for(auto j=0;j < ui->tableOrder->columnCount();j++){
                out << QString(ui->tableOrder->item(i,j)->text() +
'\t').toUtf8();
                out << QString("\n").toUtf8();
            }

            file.close();

            on_pushButtonCancelhideResponseDate_clicked();
        }
    }

    void MainWindow::on_pushButtonCancelhideResponseDate_clicked()
    {
        ui->pushButtonResponseDate->hide();
        ui->lineEdit_3->hide();
        ui->lineEdit_4->hide();
        ui->pushButtonCancelhideResponseDate->hide();

        ui->pushButton_4->show();
    }

    void MainWindow::on_pushButtonAddOrderFile_clicked()
    {
        QString num;
        ui->lineEditNumOrderForFile->text() <= '9' && ui->lineEditNumOrderForFile-
>text() >='0'?
            num = ui->lineEditNumOrderForFile->text();
            num = "Error";

        if(num.toInt()-1 >= ui->tableOrder->rowCount()) num = "Error";
    }
}

```

```

    if(num == "Error") return;

    QFile file("order №" + num+ ".doc");

    if (!file.open(QIODevice::WriteOnly | QIODevice::Text))
    {
        print "Ошибка"; return;
    }

    num = QString::number(num.toInt()-1);
    QTextStream out(&file);
    out << QString("Дата и время\tБлюда\tСчет\n").toUtf8();
    for(auto i =0; i<ui->tableOrder->rowCount();i++){
        out << ui->tableOrder->item(num.toInt(),i)->text() + '\t';
    }

    file.close();
}

```

Файл main.cpp

```

#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}

```

Формы

Файл MainWindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>MainWindow</class>
    <widget class="QMainWindow" name="MainWindow">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>1154</width>
                <height>652</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>MainWindow</string>
        </property>
        <widget class="QWidget" name="centralWidget">
            <widget class="QTabWidget" name="tabWidget">
                <property name="geometry">
                    <rect>
                        <x>0</x>
                        <y>0</y>
                        <width>1161</width>
                        <height>651</height>
                    </rect>
                </property>
            </widget>
        </widget>
    </widget>
</ui>

```

```

<property name="currentIndex">
  <number>2</number>
</property>
<widget class="QWidget" name="tab">
  <attribute name="title">
    <string>Блюда</string>
  </attribute>
  <widget class="QPushButton" name="pushButton">
    <property name="geometry">
      <rect>
        <x>510</x>
        <y>20</y>
        <width>151</width>
        <height>25</height>
      </rect>
    </property>
    <property name="text">
      <string>Составить меню</string>
    </property>
  </widget>
  <widget class="QTableWidget" name="tableDish">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>501</width>
        <height>450</height>
      </rect>
    </property>
    <column>
      <property name="text">
        <string>Название</string>
      </property>
    </column>
    <column>
      <property name="text">
        <string>Рецепт блюда</string>
      </property>
    </column>
    <column>
      <property name="text">
        <string>Цена</string>
      </property>
    </column>
    <column>
      <property name="text">
        <string>Себестоимость</string>
      </property>
    </column>
  </widget>
  <widget class="QPushButton" name="pushButtonAddDish">
    <property name="geometry">
      <rect>
        <x>510</x>
        <y>60</y>
        <width>151</width>
        <height>31</height>
      </rect>
    </property>
    <property name="text">
      <string>Добавить блюдо</string>
    </property>
  </widget>

```

```

<widget class="QWidget" name="layoutWidget_3">
  <property name="geometry">
    <rect>
      <x>680</x>
      <y>20</y>
      <width>351</width>
      <height>153</height>
    </rect>
  </property>
  <layout class="QVBoxLayout" name="LayoutAddProduct_3">
    <item>
      <widget class="QLineEdit" name="lineEditNumDish">
        <property name="text">
          <string>Название</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEditRecipeDish">
        <property name="text">
          <string>Рецепт (Продукты для приготовления, через запятую)</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEditPriceDish">
        <property name="text">
          <string>Цена</string>
        </property>
      </widget>
    </item>
    <item>
      <layout class="QHBoxLayout" name="horizontalLayout_3">
        <item>
          <widget class="QPushButton" name="pushButtonCancelDish">
            <property name="text">
              <string>Отмена</string>
            </property>
          </widget>
        </item>
        <item>
          <widget class="QPushButton" name="pushButtonSaveDish">
            <property name="text">
              <string>Сохранить</string>
            </property>
          </widget>
        </item>
      </layout>
    </item>
  </layout>
</widget>
</widget>
<widget class="QWidget" name="tab_2">
  <attribute name="title">
    <string>Заказы</string>
  </attribute>
  <widget class="QTableWidget" name="tableOrder">
    <property name="geometry">
      <rect>
        <x>10</x>
        <y>20</y>
        <width>381</width>
        <height>450</height>
      </rect>
    </property>
  </widget>
</widget>

```

```

    </rect>
  </property>
</column>
  <property name="text">
    <string>Дата </string>
  </property>
</column>
<column>
  <property name="text">
    <string>Блюда в заказе</string>
  </property>
</column>
<column>
  <property name="text">
    <string>Счет заказа</string>
  </property>
</column>
</widget>
<widget class="QPushButton" name="pushButton_4">
  <property name="geometry">
    <rect>
      <x>560</x>
      <y>30</y>
      <width>181</width>
      <height>25</height>
    </rect>
  </property>
  <property name="text">
    <string>Заказы за период</string>
  </property>
</widget>
<widget class="QPushButton" name="pushButtonAddOrder">
  <property name="geometry">
    <rect>
      <x>560</x>
      <y>60</y>
      <width>181</width>
      <height>25</height>
    </rect>
  </property>
  <property name="text">
    <string>Добавить заказ</string>
  </property>
</widget>
<widget class="QWidget" name="layoutWidget_2">
  <property name="geometry">
    <rect>
      <x>760</x>
      <y>30</y>
      <width>351</width>
      <height>101</height>
    </rect>
  </property>
  <layout class="QVBoxLayout" name="LayoutAddProduct_2">
    <item>
      <widget class="QLineEdit" name="lineEditDateOrder">
        <property name="text">
          <string>Дата (пример 2019.01.01) </string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEditDIshOrder">

```

```

    <property name="text">
      <string>Блюда через запятую</string>
    </property>
  </widget>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_2">
    <item>
      <widget class="QPushButton" name="pushButtonCancelOrder">
        <property name="text">
          <string>Отмена</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QPushButton" name="pushButtonSaveOrder">
        <property name="text">
          <string>Сохранить</string>
        </property>
      </widget>
    </item>
  </layout>
</item>
</layout>
</widget>
<widget class="QWidget" name="verticalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>760</x>
      <y>140</y>
      <width>291</width>
      <height>121</height>
    </rect>
  </property>
  <layout class="QVBoxLayout" name="verticalLayout_4">
    <item>
      <widget class="QLineEdit" name="lineEdit_3">
        <property name="text">
          <string>Начальная дата (пример 2019.01.01) </string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit_4">
        <property name="text">
          <string>Конечная дата</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QPushButton" name="pushButtonResponseDate">
        <property name="text">
          <string>Составить отчет</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QPushButton" name="pushButtonCancelhideResponseDate">
        <property name="text">
          <string>Отмена</string>
        </property>
      </widget>
    </item>
  </layout>
</widget>

```

```

</layout>
</widget>
<widget class="QWidget" name="verticalLayoutWidget_2">
  <property name="geometry">
    <rect>
      <x>560</x>
      <y>100</y>
      <width>181</width>
      <height>80</height>
    </rect>
  </property>
  <layout class="QVBoxLayout" name="verticalLayout">
    <item>
      <widget class="QLineEdit" name="lineEditNumOrderForFile">
        <property name="text">
          <string>Номер заказа</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QPushButton" name="pushButtonAddOrderFile">
        <property name="text">
          <string>Сохранить заказ в отчет</string>
        </property>
      </widget>
    </item>
  </layout>
</widget>
</widget>
<widget class="QWidget" name="tab_5">
  <attribute name="title">
    <string>Продукты</string>
  </attribute>
  <widget class="QPushButton" name="pushButtonProductFile">
    <property name="geometry">
      <rect>
        <x>470</x>
        <y>20</y>
        <width>151</width>
        <height>25</height>
      </rect>
    </property>
    <property name="text">
      <string>Составить отчет</string>
    </property>
  </widget>
  <widget class="QPushButton" name="pushButtonAddPosition">
    <property name="geometry">
      <rect>
        <x>470</x>
        <y>60</y>
        <width>151</width>
        <height>25</height>
      </rect>
    </property>
    <property name="text">
      <string>Добавить позицию</string>
    </property>
  </widget>
  <widget class="QTableWidget" name="tableProduct">
    <property name="geometry">
      <rect>
        <x>0</x>

```

```

        <y>0</y>
        <width>381</width>
        <height>450</height>
    </rect>
</property>
</column>
<column>
    <property name="text">
        <string>Название</string>
    </property>
</column>
<column>
    <property name="text">
        <string>Кол-во на складе</string>
    </property>
</column>
<column>
    <property name="text">
        <string>Стоимость</string>
    </property>
</column>
</widget>
<widget class="QWidget" name="layoutWidget">
    <property name="geometry">
        <rect>
            <x>650</x>
            <y>20</y>
            <width>351</width>
            <height>122</height>
        </rect>
    </property>
    <layout class="QVBoxLayout" name="LayoutAddProduct">
        <item>
            <widget class="QLineEdit" name="lineEditName">
                <property name="text">
                    <string>Название продукта</string>
                </property>
            </widget>
        </item>
        <item>
            <widget class="QLineEdit" name="lineEditCount">
                <property name="text">
                    <string>Кол-во на складе</string>
                </property>
            </widget>
        </item>
        <item>
            <widget class="QLineEdit" name="lineEditPrice">
                <property name="text">
                    <string>Себестоимость</string>
                </property>
            </widget>
        </item>
        <item>
            <layout class="QHBoxLayout" name="horizontalLayout">
                <item>
                    <widget class="QPushButton" name="pushButtonCancel">
                        <property name="text">
                            <string>Отмена</string>
                        </property>
                    </widget>
                </item>
                <item>
                    <widget class="QPushButton" name="pushButtonSave">

```



```
        <property name="text">
            <string>Сохранить</string>
        </property>
    </widget>
</item>
</layout>
</item>
</layout>
</widget>
</widget>
</widget>
</widget>
</widget>
</widget>
<layoutdefault spacing="6" margin="11"/>
<resources/>
<connections/>
</ui>
```