**Министерство образования и науки российской федерации**

ЮГОРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровой экономики

**Отчет по Учебной (ознакомительной) практике**

Место прохождения практики: ЮГОРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Выполнил: студент 2 курса группы 1191б

Петров Владислав Вячеславович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Руководитель практики:

Русанов Михаил Александрович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Ханты-Мансийск – 2021

Содержание

Оглавление

[**Введение** 3](#_Toc68983993)

[**Раздел 1** 4](#_Toc68983994)

[**Раздел 2** 5](#_Toc68983995)

[**Раздел 3** 8](#_Toc68983996)

[**Раздел 4** 11](#_Toc68983997)

[**Вывод** 14](#_Toc68983998)

[**Список литературы** 15](#_Toc68983999)

[**Приложение** 16](#_Toc68984000)

# **Введение**

Ежедневник - деловой блокнот, предназначенный для ежедневного ведения записей о делах, планах, результатах и прочих записях, связанных с ежедневной деловой деятельностью человека или компании/организации. Каждый человек ежедневно выполняет широкий спектр задач/дел, и порой такое приложение как ежедневник может очень сильно сэкономить время и человеческие ресурсы. Мы все любим тайм-менеджмент и когда весь день расписан задачами – становится намного проще следить за временем и планировать день заранее, благодаря таким приложениям успешные или известные люди очень часто оказываются в плюсе, так как они экономят время и силы. Я буду делать свое приложение ежедневник с помощью платформы Windows Forms, используя Базу Данных для хранения информации, а связующее звено будет DataGridView и DataBinding, с помощью свойств которых мы можем очень просто и что самое главное быстро обращаться к БД из-под формы без потери данных и с возможностью их изменения, фильтрации и т.д. Платформа WF была выбрана только потому что можно красиво и быстро руководить пространством, а так же управлять данными.

# **Раздел 1**

**Постановка задач. Проектирование.**

* 1. Постановка задач.

1. Выбор способа подключения Базы Данных к Форме.
2. Построение дружелюбного графического интерфейса для пользователя с помощью компонентов Windows Forms.
3. Создание Системы Контроля Версий на GitHub.
4. Реализация стандартных функций с помощью компонентов WF и командлет MSSQL:
   * + 1. INSERT (Ввод данных)
       2. UPDATE (Обновление данных)
       3. DELETE (Удаление данных)
5. Добавление функций сортировки/фильтрации и поиска данных.
6. Создание защиты от неопытного пользователя в программу
   1. Проектирование.

После постановки задач – начинаем проектировать наш интерфейс и его удобство.

1. Поверх формы наложить DataGridView для отображения и дальнейших операций с данными.
2. Реализовать каждую функцию на отдельной странице (форме или табконтроле).
3. Для каждой функции добавить нужные поля для ввода и управления данными.
4. Сделать отдельный элемент для отображения ошибок.
5. Сделать выравнивание элементов для более эстетичного вида.
6. Ограничить доступ к вводу для защиты от неопытного пользователя, а также обработчик ошибок для исключительных случае.

# **Раздел 2**

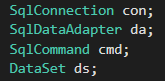
**Программный код. Основные функции.**

С самого начала подключаем пространство имен для последующей работы с Базой Данных (Листининг.1):



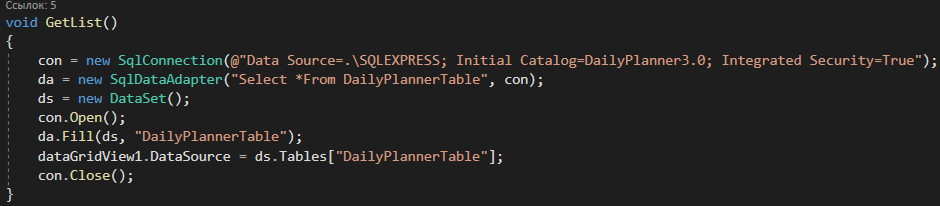
Листининг.1

Далее объявляем переменные для подключения Базы Данных и управлением данными на форме (Листининг.2):



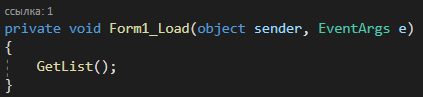
Листининг.2

После создаем метод для постоянного обновления данных после выполнения каждой функции, а также связи между Базой Данных и Windows Form (Листининг.3):



Листининг.3

Теперь можем заполнить ранее созданный DataGridView (Листининг.4):



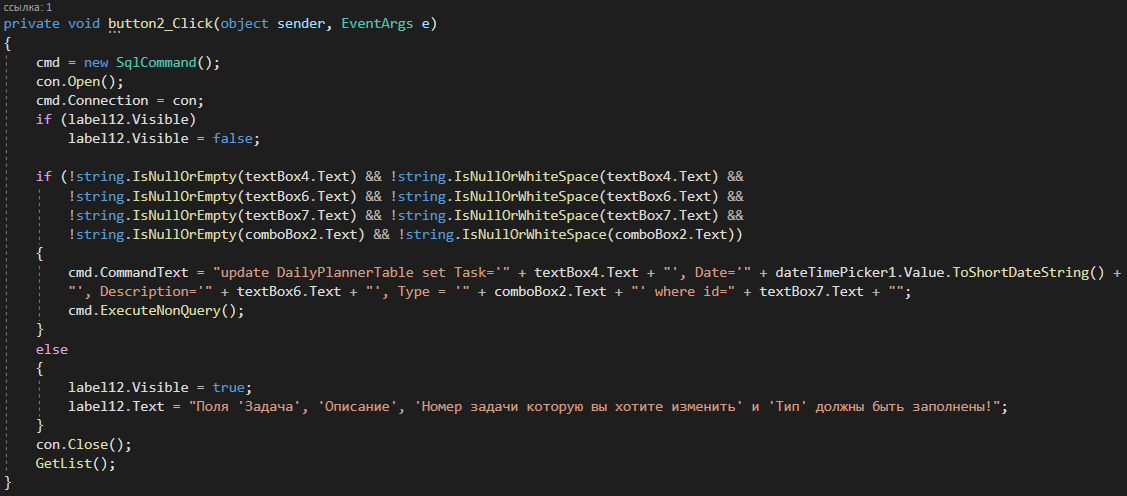
Листининг.4

Реализуем первую функцию INSERT с помощью передачи текста из компонентов формы в таблицу данных и моментальное обновление с помощью метода обновления GetList (), так же присутствует защита от неопытного пользователя и обработка исключений (Листининг.5):



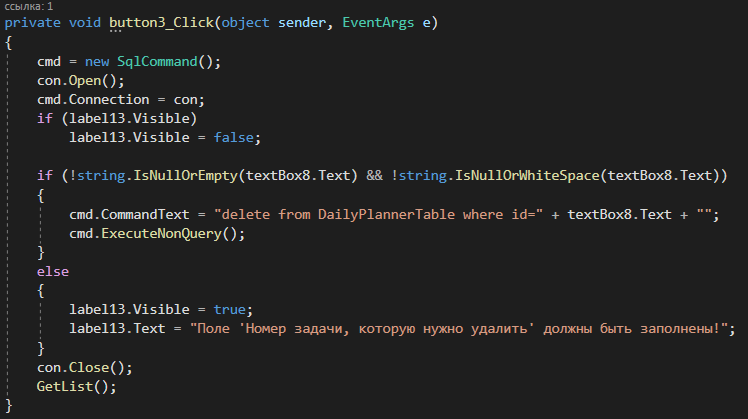
Листининг.5

После чего подобным образом добавляем еще одну функцию UPDATE, командлета SQL немного изменилась, и теперь напрямую обращаемся к PrimaryKey, т.е. полю ID (Листининг.6):



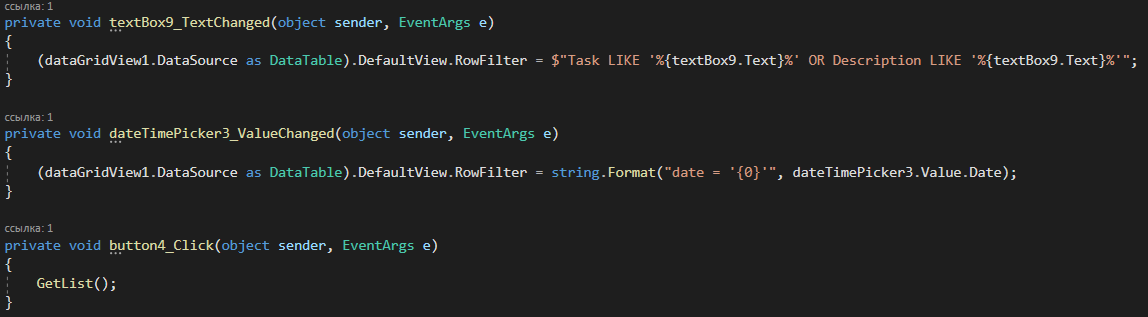
Листининг.6

А также функцию DELETE, которая удаляет всю строку из Базы Данных по идентификатору т.е. полю ID (Листининг.7):



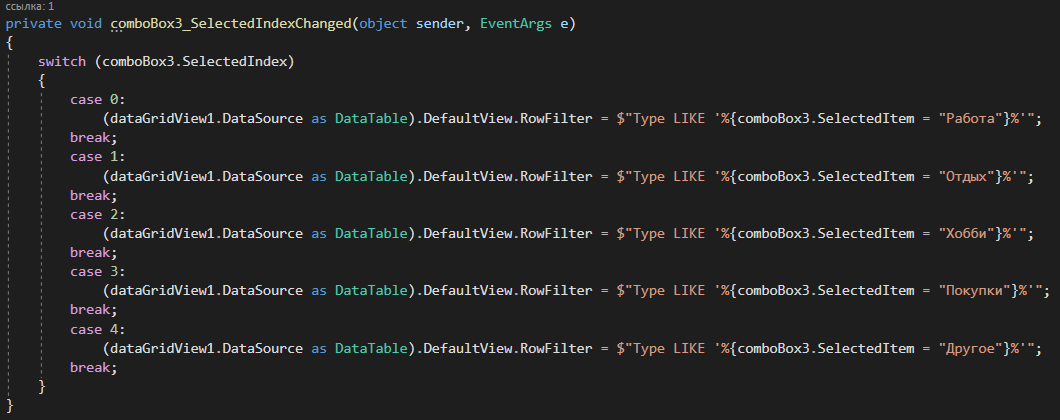
Листининг.7

Далее добавляем поиск по тексту и фильтрацию с помощью DateTimePicker, и отдельный управляющий элемент для сброса фильтрации (Листининг.8):



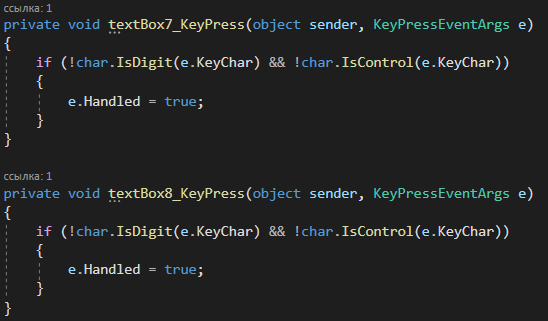
Листининг.8

Типизацация выполняется с помощью отдельного управляющего элемент ComboBox, который включает в себя некоторое количество вариантов типа (Листининг.9):



Листининг.9

Защита от неопытного пользователя выполняется не только выделением сообщением о не заполненности поля или обработкой исключений, но еще и ограничением в действиях пользователя запретом на ввод некоторых символов (Листининг.10):



Листининг.10

Выход из программы можно выполнить не только с помощью стандартных органов формы, но и с помощью ToolStrip (Приложение.А.1).

# **Раздел 3**

**Графический интерфейс**

Графический интерфейс интуитивно понятен. Он включает в себя ToolStrip для выхода из программы, TabControl для переключения между функциями, а также DataGridView для отображения таблицы данных из Базы Данных. Для передвижения по нужным функциям нужно просто переключить вкладку на нужную с помощью TabControl, и начинать заполнять данные в TextBox, ComboBox и DateTimePicker, после чего нужно нажать на орган управление в виде Button для изменения данных в DataGridView и базе данных.

Самая первая вкладка интерфейса – Отображение, является пустой версией TabPage для просмотра данных (Рисунок.1):

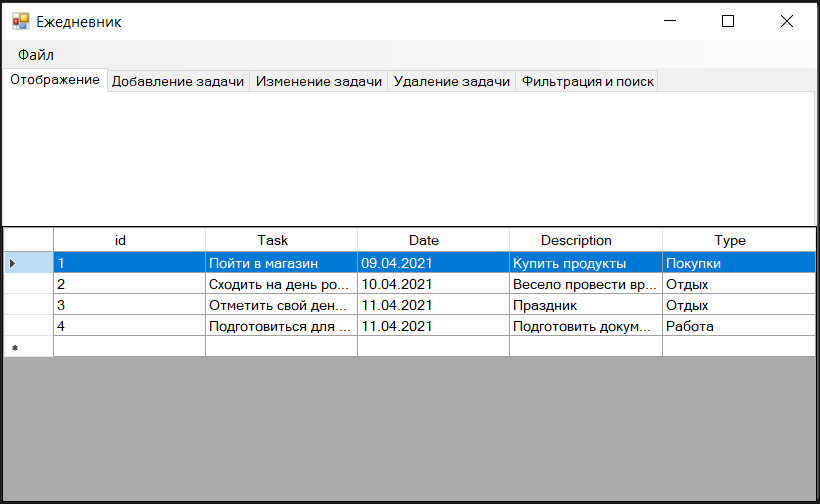


Рисунок.1

Следующая вкладка представляет собой реализацию функции вставки данных, есть поля для в вода текста и поля выбора, к которым прикреплены описательные элементы, указывающие что нужно сделать в полях ввода (Рисунок.2):

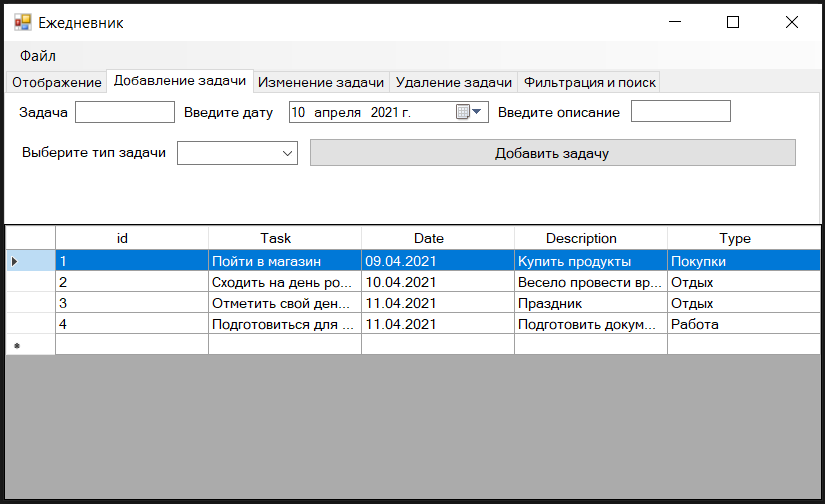


Рисунок.2

После вкладки вставка идет Изменение данных, оно выполняется аналогично прошлому, нужно просто ввести данные для изменения, смотря на надписи около полей ввода, а далее нажать на кнопку для внесения изменений в таблицу данных (Рисунок.3):

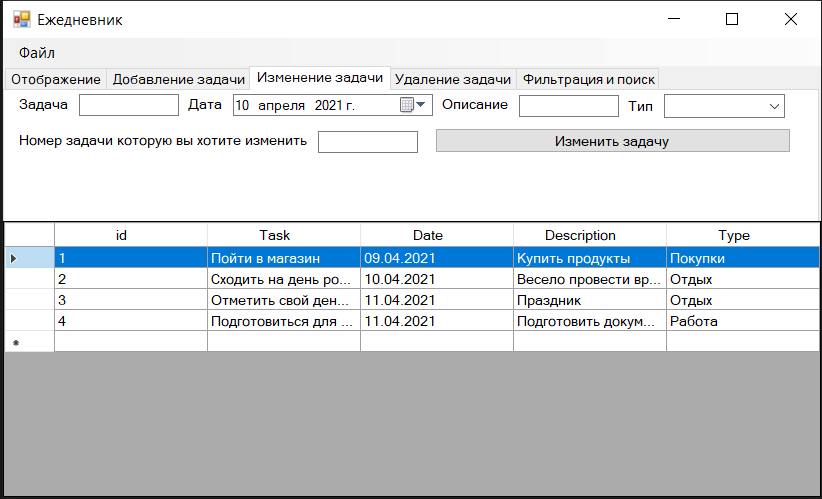


Рисунок.3

Далее следует вкладка Удаления данных из таблиц и работает благодаря полю идентификатора ID в базе денных, элементы взаимодействия не изменяются (Рисунок.4):

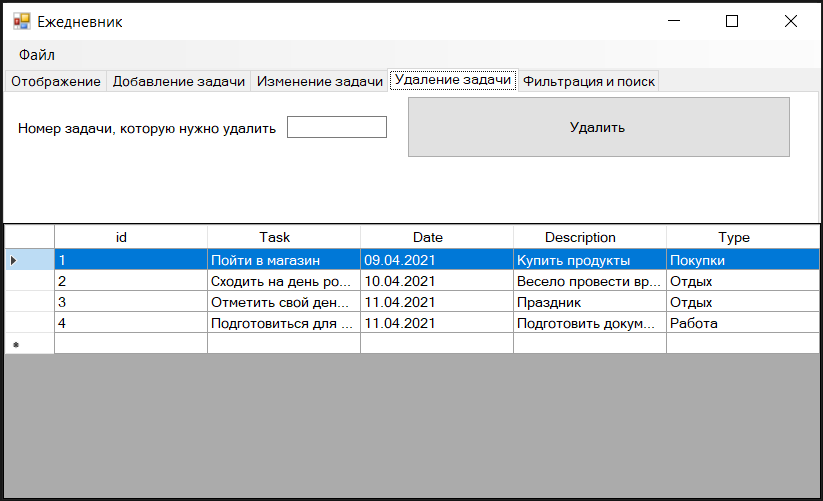


Рисунок.4

Последняя вкладка интерфейса реализует сразу несколько функций, это: Поиск, Типизация и Фильтрация, все функции используют те же элементы что и прошлые вкладки, а также присутствует кнопка обнуления фильтрации (Рисунок.5):

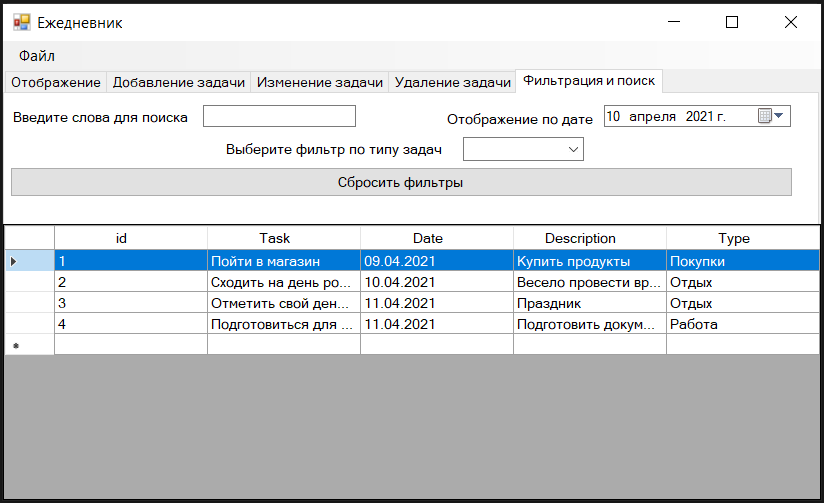


Рисунок.5

# **Раздел 4**

**Тестирование программы.**

Тестирование программы начинается с попытки получить исключительный случай в полях взаимодействия, ввод пустых данных во всех вкладках по очереди, но так как в коде прописана защита от неопытного пользователя – ввести пустые поля не является возможным и будет отображено уведомление что было сделано не так (Рисунок.6):

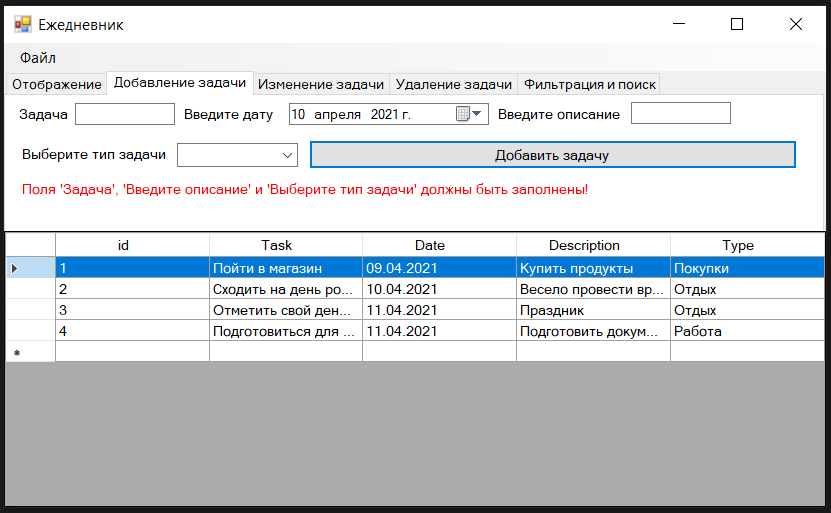


Рисунок.6

Теперь попробуем ввести в поля, где вводятся строгие значения (такие как ID), ввести специальные символы или буквы, но, так как стоит ограничение на вводимые символы на таких полях, то они будут игнорироваться и ничего кроме цифр не будет введено (Рисунок.7):

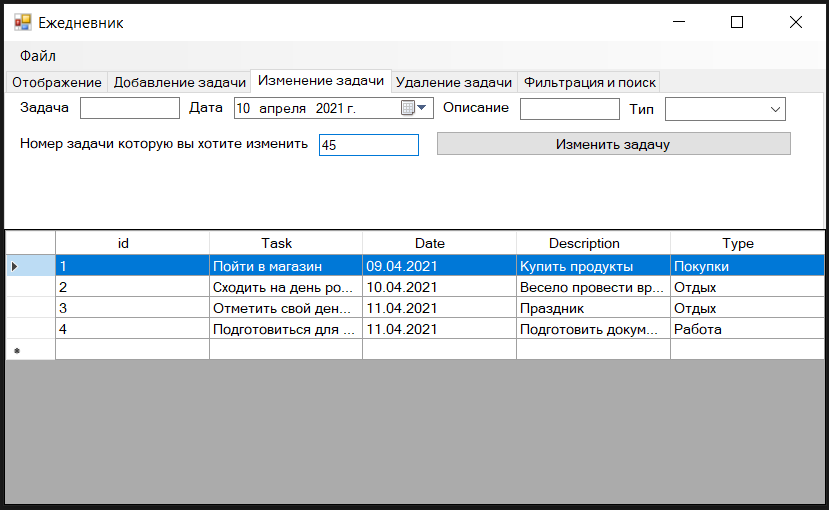


Рисунок.7

Далее проверим работают ли функции в целом, начиная со вставки (Рисунок.8):

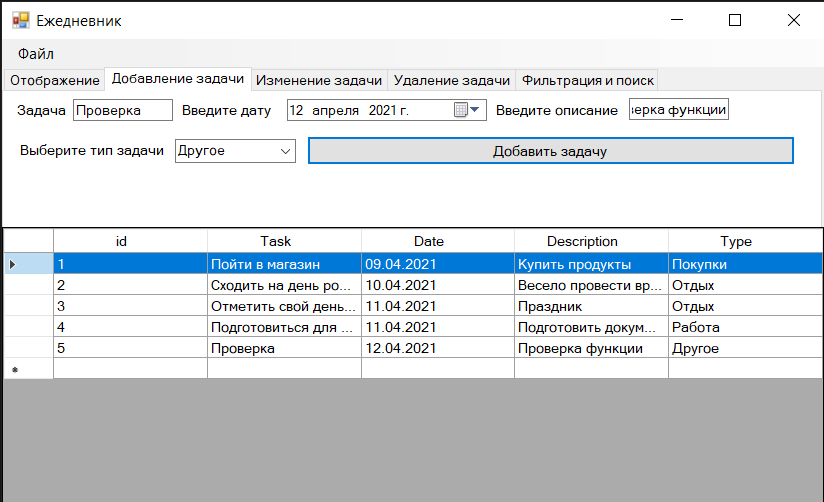


Рисунок.8

Поиск данных осуществляется с помощью User-friendly элементов в которых невозможно допустить ошибку, а если она и произойдет, то работа программы прекратиться благодаря обработчику исключительных случаев и данные не будут утеряны, а данные в обычном поле TextBox сортируются благодаря TextChangedIvent что исключает ошибки как таковые. Далее проверяем правильность работы поиска (Рисунок.9):

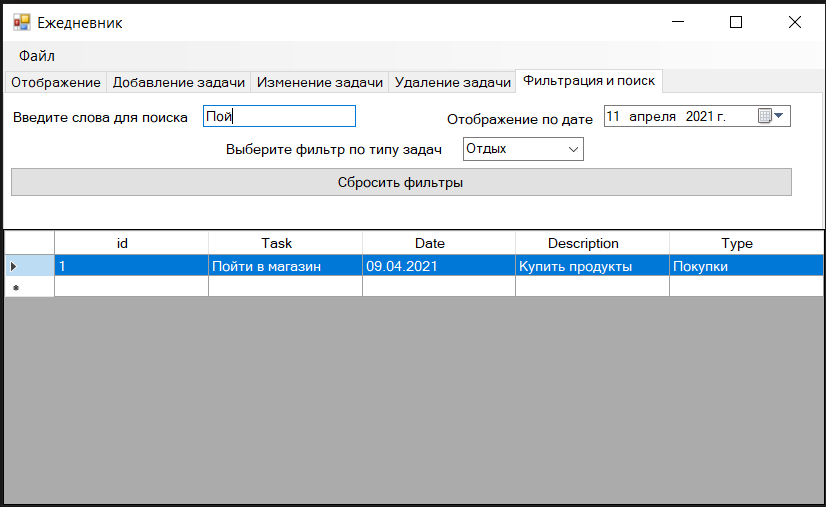


Рисунок.9

И в завершении проверим как проходит выход из приложения через ToolStrip (Рисунок.10):



Рисунок.10

Так мы видим, что она полностью завершает свою работу и автоматически отключает не просто графический интерфейс, но и полностью выходит из программы.

# **Вывод**

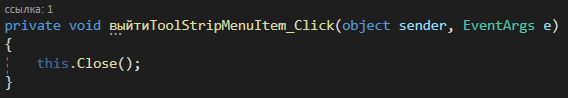
Программа имеет понятный и удобный интерфейс благодаря средствам Windows Forms, так же добавлена защита от неопытного пользователя и обработчик исключений, сама программа выполняет все нужные функции. Вся программа опубликована в системе контроля версий GitHub. Хранилище данных реализовано отдельным файлом базы данных в формате .mdf в папке с программой.

# **Список литературы**

1. Книга-руководство по языку программирования C#: <http://ijevanlib.ysu.am/wp-content/uploads/2017/12/C-4.0-полное-руководство-Герберт-Шилдт_2011.pdf>
2. Книга для продвинутых пользователей по C# и .NET FRAMEWORK: <https://viduus.net/wp-content/uploads/2018/02/Rihter-Dzh.-CLR-via-C.-Programmirovanie-na-platforme-Microsoft-.NET-Framework-4.5-na-yazyke-C-Master-klass-2013.pdf>

# **Приложение**

Приложение.А:



Приложение.A.1.