**ВВЕДЕНИЕ**

На данный момент все больше и больше развиваются и улучшаются технологии, много новых приложений появляется с каждым днем, и никому точно не известно, что они из себя представляют. Скачивая файл с нужной программой или какой-то полезной информацией нельзя быть уверенным на все 100%, что этот файл безопасный. Разумеется, для этого придумали антивирусы, но не все антивирусы работают хорошо. Приложение, позволяющее искать некоторый набор запрещенных слов в файлах может помочь в поиске вредоносных файлов.

Им также можно пользоваться, когда требуется найти файл с определенным набором слов в определенной папке. Данное приложение позволяет отыскать тот или иной файл с содержанием тех или иных наборов слов, просто выбрав нужные слова и кликнув один раз на кнопку «Поиск». Это приложение как Google, но только на локальных файлах. Данное приложение отображает полный путь к файлу, количество искомых слов, общую статистику за все время поиска, топ самых искомых слов и т.д. Оно может применяться как дома, так и в офисах, причем на разных компьютерах, так как само считывает содержимые файлы и папке на компьютере, на котором было запущено.

Интерфейс приложения достаточно легок и понятен, поэтому для каждого пользователя не составит труда разобраться в нем.

**1 ПОСТАНОВКА ЗАДАЧИ**

Цель данной работы – создание приложения, которое будет позволять искать некоторый набор запрещенных слов в файлах с применением Microsoft.Net Framework используя интерфейс программирования Windows Forms.

Перед разработкой данного приложения были поставлены следующие задачи:

* Пользовательский интерфейс приложения должен позволять ввести или загрузить из файла набор запрещенных слов.
* При нажатии на кнопку «Старт», приложение должно начать искать эти слова на всех доступных накопителях информации (жесткие диски, флешки).
* Файлы, содержащие запрещенные слова, должны быть скопированы в заданную папку.
* Кроме оригинального файла, нужно создать новый файл с содержимым оригинального файла, в котором запрещенные слова заменены на 7 повторяющихся звезд (\*\*\*\*\*\*\*).
* Также нужно создать файл отчета. Он должен содержать информацию о всех найденных файлах с запрещенными словами, пути к этим файлам, размер файлов, информацию о количестве замен и так далее.
* В файле отчета нужно также отобразить топ самых популярных запрещенных слов.
* Интерфейс программы должен показывать прогресс работы приложения с помощью индикаторов (progress bars).
* По итогам работы программы необходимо вывести результаты работы в элементы пользовательского интерфейса (нужно продумать, какие элементы управления понадобятся).
* Программа обязательно должна использовать механизмы многопоточности и синхронизации! Программа может быть запущена только в одной копии.

**2 ТРЕБОВАНИЯ К ПРОГРАМНОМУ ОБЕСПЕЧЕНИЮ**

**2.1 Требование к функциональным характеристикам системы**

Программа должна обеспечить возможность выполнения следующих функций:

- получения доступа ко всем директориям данного компъютера и содержащих файлов;

- получения доступа к MS SQLServer для создания базы данных на компътере (заносится статистика искомых слов) и сохранение данных в БД;

- правильная настройка подключения MS SQLServer на компъютере

**2.2 Требования к составу и параметров технических средств**

Программа должна работать на на базе операционной системы Windows.

Минимальная конфигурация:

Тип процессора: IntelCore i3 и выше.

Объем оперативной памяти: 8Гб и више.

**2.3 Требование к програмным характеристикам**

Для корректной работы программы, машины, на которой будет происходить ее запуск должны отвечать следующим программным требованиям:

* операционная система Windows 10;
* платформа – 64х.
* .Net версия 4.7.2

**3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ**

**3.1 Проектирование интерфейса**

При запуске приложения открывается сразу главное окно, в котором находятся такие компоненты, как:

- TreeView – для отображения всех директорий, которые находятся на данном компьютере. Оно заполняется сразу при открытии приложения в отдельном потоке, чтобы форма самого приложения не подвисала.

- ListBox – на котором находятся все найденные файлы (точнее полные пути к файлам) по заданным словам. Если файлы найдены не будут на ListBox ничего не отобразится.

- Три главные кнопки:

* Save in… - используется для копирования всех найденных файлов в отдельную папку. При наведении курсором на данную кнопку, под ней отображается текущий путь для сохранения. При нажатии на кнопку можно изменить этот текущий путь, выбрав в TreeView другую папку. Чтобы указать функционал пользователю, к этой кнопке был приклеплен ToolTip (подсказка), с текстом «Select a folder and click», таким образом делая приложение легкое для понимания.
* Choose Words – кнопка, которая открывает новую форму Words для того, чтобы задать искомые слова.
* и главная кнопка Search, которая отвечает за сам поиск слов. Функция по поиску слов работает в другом потоке, для того, чтобы форма не подвисала и пользователю было комфортно работать в этом приложении.

После нажатия на данную кнопку появляются такие компоненты, как ProgressBar – который показывает текущее выполнение поиска заданных слов в файле и Lable Files, который показывает, какие файлы на данный момент проверяются компьютером (их полный путь).

После того, как приложение закончит поиск, при условии, если файлы будут найдены и ListBox не будет пустым, появляется еще один компонент такой, как кнопка Show Statistic, которая вызывает новую форму Show Statistic.

**3.1.1 Интерфейс формы Words**

Форма Words состоит из таких компонентов, как:

- TextBox, в который прописывается то запрещенное слово, которое пользователь хотел бы найти

-кнопка «+» которая добавляет вписанное слово в ListBox

-кнопка «-» которая удаляет выбранное слово с ListBox

- кнопка с изображением файла, которая загружает текстовый файл в ListBox с запрещенными словами.

Таким образом пользователь может не только добавлять слова вручную, но и загружать весь файл со словами, так же ему предоставляется возможность добавить вручную слово, после загрузки его через файл

* сам ListBox, в котором и храняться все слова, который пользователь добавляет.

**3.1.2 Интерфейс формы Words Show Statistic**

На данной форме находится только два компонента:

- DataGridView, в который передается статистика найденных слов, а именно само слово, путь к файлу, в котором оно было найдено, размер этого файла, количество слов, которое было найдено и количество слов, которое было изменено на «\*\*\*\*\*\*\*» (как было сказано в задании).

- кпонка, которая выводит Топ самых популярных запрещенных слов.

Данная кнопка открывает еще одну форму Top Words.

**3.1.2.1 Интерфейс формы Top Words**

Интерфейс данной формы очень прост, он имеет только один компонент DataGridView который отображает Топ всех запрещенных слов (а именно само слово и количество раз, которое оно было найдено). Данная статистика дает понять пользователю, какие слова более часто используются в поиске.

**3.2 Проектирование диаграммы классов**

Диаграмма классов ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Static Structure diagram) — структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)) языка моделирования [UML](https://ru.wikipedia.org/wiki/UML), демонстрирующая общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) системы, их коопераций, [атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0) (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Так как приложение написано на Windows Forms, главным и первым классом будет класс Form.

Диаграмму классов данного приложения выглядит следующим образом (рисунок 3.2.1 и рисунок 3.2.2).

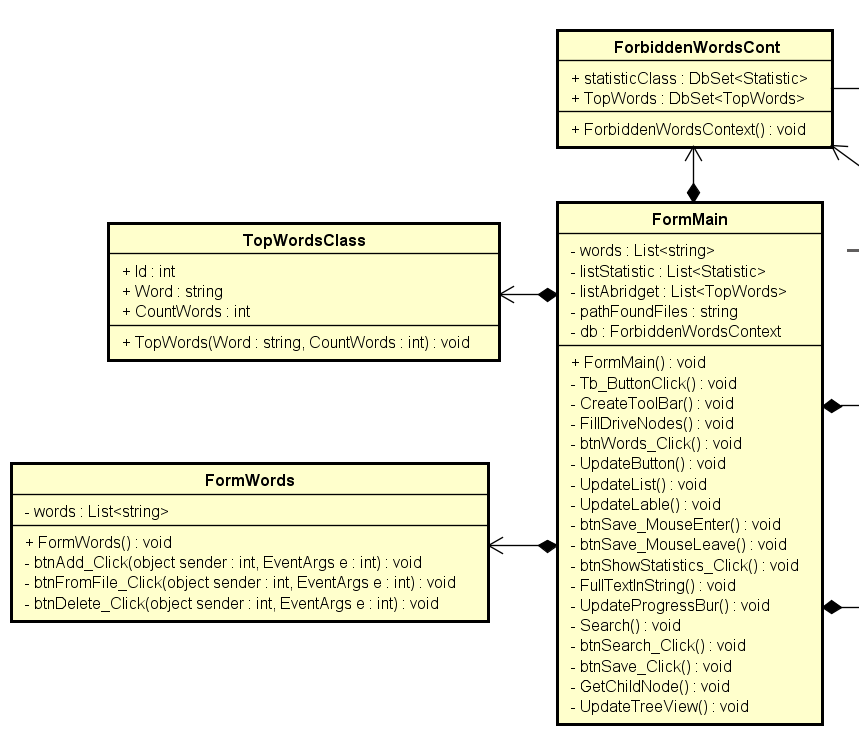


Рисунок 3.2.1 Диаграмма классов

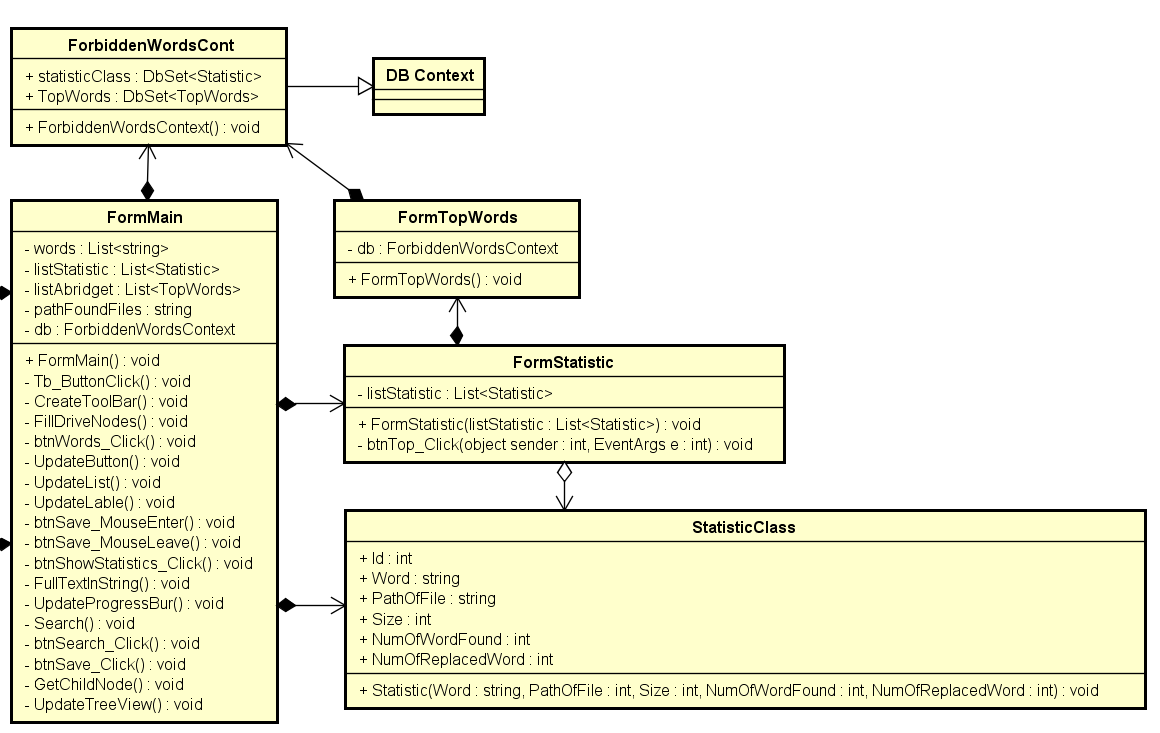


Рисунок 3.2.2 Диаграмма классов

Как видно с данных рисунков, вся программа состоит из 7 классов, которые тесно взаимосвязаны друг с другом.

Класс Form (FormMain) – это основной класс, в котором содержится основной код программы. В ней вызываются такие классы, как

* ForbiddenWordsContext (который наследуется от класса DB Context)– класс, отвечающий за подключение к базе данных и считывание / запись информации в нее. Этот класс содержит два публичных свойства типа DBSet<> тех классов, которые используются для заполнения таблиц базы данных, а именно Statistic и TopWords.
* класс Statistic отвечает за передачу данных в базу данных в таблицу Statistic. В нем содержаться все необходимые публичные свойства для получения информации найденного файла, а именно само слово, путь к файлу, в котором найдено слово, размер этого файла, количество замененных и найденных слов. Также в нем содержится конструктор, через который можно передавать данные параметры.
* класс TopWords который также отвечает за передачу данных в базу данных в другую таблицу TopWords. В нем содержаться все необходимые публичные свойства для получения информации о найденных словах (само слово и количество этих слов). Также в нем содержится конструктор, через который можно передавать данные параметры.
* FormWords - который отвечает за то, чтобы задавать нужные слова для поиска. В ней содержится список искомых слов (передаваемый с главной формы), так как тип ссылочный, он инициализируется сразу при добавлении слова в ListBox на данной форме. Также на данной форме обработчики событий, которые отвечают за добавление, удаление или загрузку нужных слов.
* FormStatistic – в данный класс с главной формы в конструктор класса передается лист класса Statistic, в котором и содержится вся информация о найденных файлах. Также в данном классе прописан обработчик события, который вызывает еще один класс FormTopWords.
* Класс FormTopWords содержит объект класса ForbiddenWordsContext, который позволяет подключится к базе данных и получить все необходимые данные, а именно с помощью функций расширения вывести в DataGridView информацию о слове и количестве найденных его за все время.

**3.3 Проектирование базы данных**

База данных (БД) - это совокупность данных, связанных между собой определенную тематику. Эти данные хранятся на машинных носителях в упорядоченном виде. Наряду с понятием баз данных существует понятие системы управления базами данных (СУБД) - это компьютерная программа, которая позволяет управлять базой данных, то есть создавать и работать с ней.

В этом приложении работа ведется с СУБД Microsoft SQL Server. Для удобного управление базами данных и разнообразными настройки в MS SQL Server существует специальная утилита SQL Server Management Studio (SSMS). Данную утилиту можно использовать для создания баз данных, таблиц, написание и выполнение запросов к БД и т.п.

В данной работе база данных создавалась с помощью кода на C#, а именно технологией Entity Framework. Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Для способа взаимодействия с базой данных был выбран Database first, с помощью которого Entity Framework создает набор классов, которые отражают модель конкретной базы данных.

При первом запуске приложения создается база данных ForbiddenWords (если она не была до этого создана или была изменена).

Таблицы база данных представлена на рисунке 3.3.1

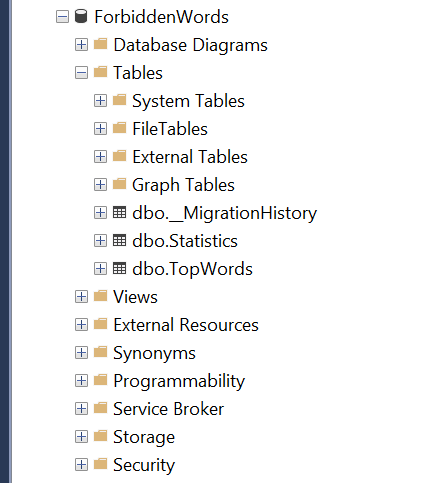


Рисунок 3.3.1 Таблицы базы данных

Соответствующие таблицам классы в Windows Form показаны на рисунке 3.3.2 и 3.3.3

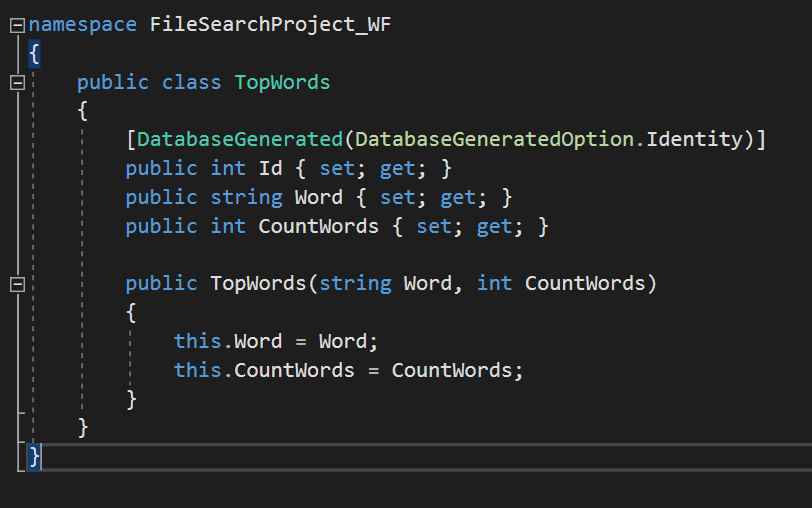


Рисунок 3.3.2 Класс TopWords

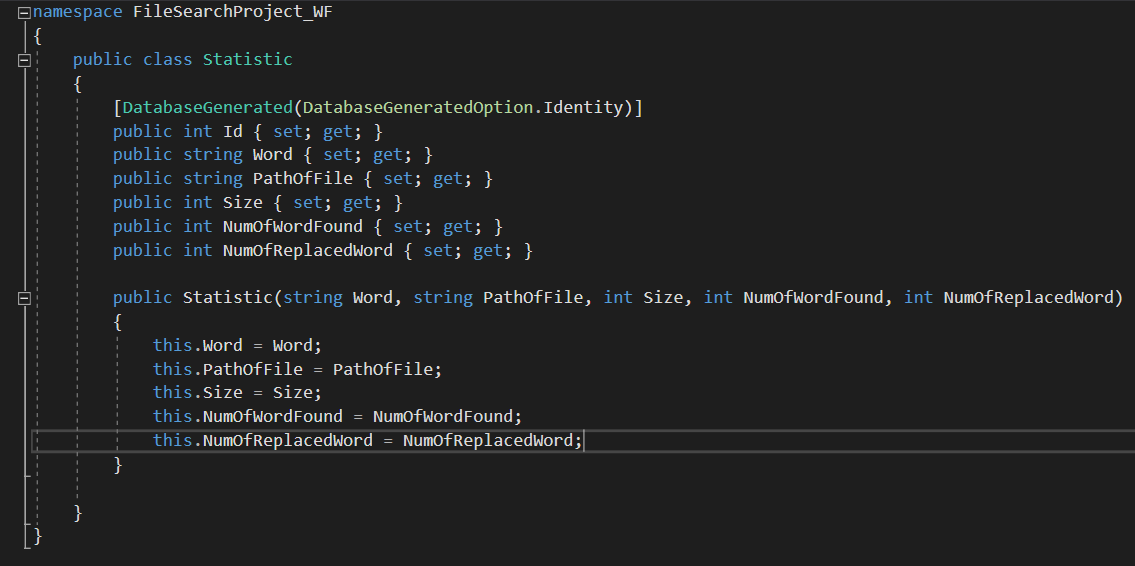


Рисунок 3.3.3 Класс Statistic

Также стоит упомянуть и о классе ForbiddenWordsContext, через который приложение может получить соответствующие данные с БД (рисунок 3.3.4)

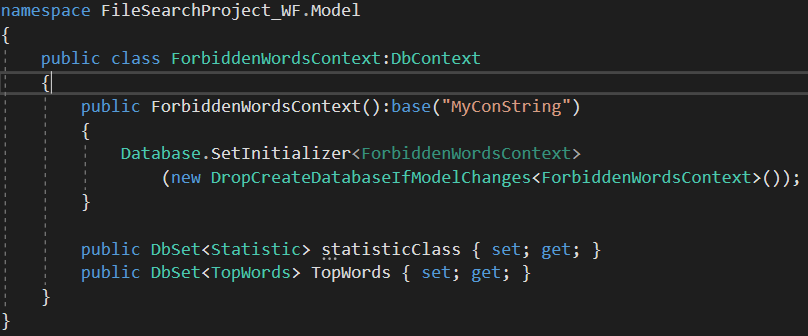


Рисунок 3.3.4 Класс ForbiddenWordsContext

Полезные данные, которые возможно получить из приложения, являются следующие:

- В таблице Statistic прописано само слово, которое было найдено, путь к файлу, в котором было найдено слово, размер самого файла, количество найденных слов в файле и количество замененных слов в файле на «\*\*\*\*\*\*\*»

-В таблице Top Words прописано само слово и количество раз за все время, сколько оно было найдено.

Таким образом, база данных хранит всю необходимую информацию по поиску запрещенных слов.

**3.4 Проектирование алгоритмов**

В данном приложении по поиску запрещенных слов самым основным алгоритмом является сам поиск. Он заключается в том, что каждый раз при запуске программы и выбора определенного диска, в котором и будет осуществляться поиск, приложение будет открывать каждый файл в каждой папке и считывать информацию, которая там находиться. Для того, чтобы осуществлять поиск, нужно выбрать те слова, которые будут ключом поиска. Более детально алгоритм можно описать следующим образом (рисунок 3.4.1 «Диаграмма деятельности»).

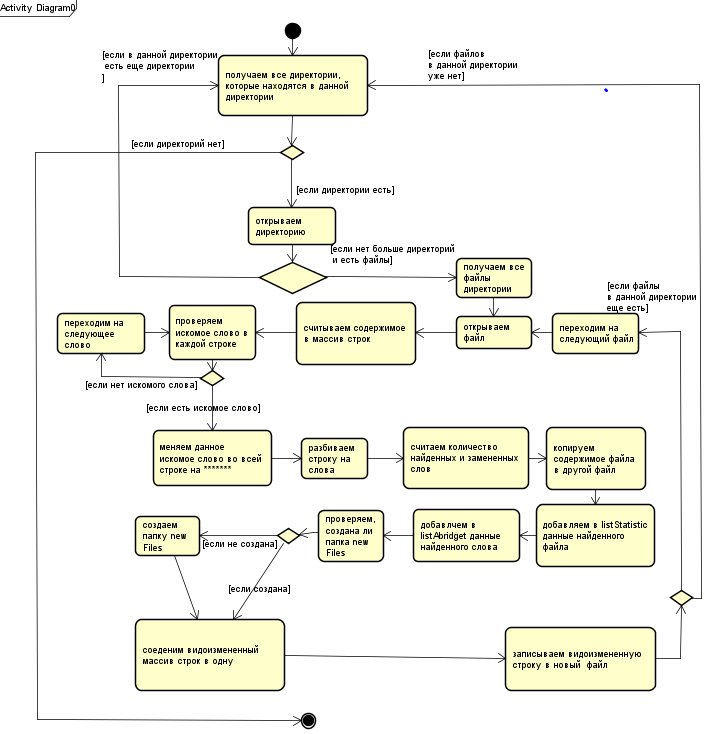


Рисунок 3.4.1 Диаграмма деятельности

1. для начала получаем все директории, которые находятся на данном выбранном диске и проходимся по каждой из них рекурсивно
2. проверяем есть ли файлы в каждом из папок
3. если файлы имеются, проходимся по каждому файлу и открываем его
4. считываем все файлы поочередно, которые находятся в данной папке, в массив строк. Это делается для того, чтобы работать не с самим файлов, постоянно подключаясь к нему, а уже с массивом строк, так как в задании сказано не только считать, но и найти нужные слова, перезаписать сами файлы в отдельную папку, заменить их на \*\*\*\*\*\*\* и так же записать замененные файлы в отдельную папку. Таким образом к данному файлу мы обращаемся только один раз.
5. Перебираем каждое запрещенное слово циклом и проверяем с помощью Regex.IsMach(), если слово в данной строке есть, заменяем каждое найденное слово на \*\*\*\*\*\*\*.
6. Если данное условие соответствует истине, тогда разбиваем данную строку на слова, при помощи функции Split
7. Считаем количество найденных и замененных слов при помощи методов расширения и в переменную IsFind передаем значение true, для того, чтобы понимать последующий ход действий.
8. Далее, если все же в файле есть искомые слова (тоесть переменная IsFind равна true) мы

- помещаем полный адрес найденного файла в ListBox (который находится на главной форме), таким образом указывая пользователю о том, что файл найден и его полный путь.

- копируем данный файл в другой файл с заранее указанным путем

- помещаем всю информацию файла в заранее созданный список, для того, чтобы далее его передать на форму Statistic и записать найденный данные в базу данных

- создаем новую папку (если таковой нет) для записи новых видоизмененных файлов (\*\*\*\*\*\*\* вместо найденных слов)

- создаем новый поток FileStrem для записи видоизмененного массива строк (заранее снова собрав его в одну цельную строку) и записываем их в файл.

Таким образом, за счёт того, что мы подключаемся к файлу только один раз и делаем все операции за раз в одном цикле, мы экономим время и ресурсы.

**4 ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММЫ**

Отладка программы - этап в разработке программы, который состоит из выявления и устранения программных ошибок, факт наличия которых все установлен.

В каждой современной системе программирования существует специальное средство для отладки программы - отладчик, который позволяет в режиме интерпретации установить контрольные точки, выполнить определенные участки программы и пересмотреть результаты работы методов. Среда разработки Visual Studio 2019, в котором была создана программа, также имеет такое средство.

Протестируем наше приложение.

Для начала зададим путь к файлу и искомые слова (рисунок 4.1)

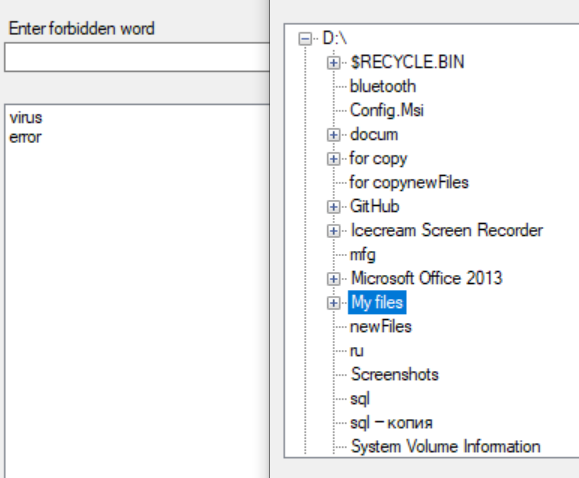


Рисунок 4.1 Искомые слова и выбор папки для поиска

И нажмем кнопку поиска.

Поставим точку остановки и запустим отладчик, убедимся в том, что функция поиска работает, находит слова, меняет их на \*\*\*\*\*\*\* (рисунок 4.2).

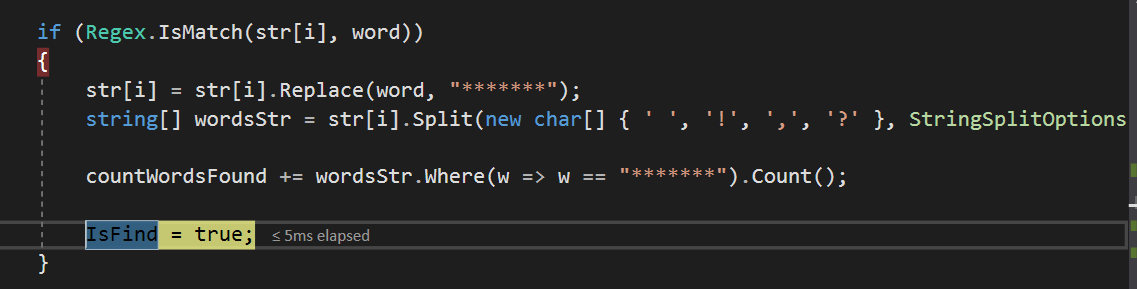


Рисунок 4.2 Часть функции Search

Данный рисунок показывает, что действительно искомое слово в функции было найдено, Regex.IsMatch() вернул значение true и все операции по подсчету и замены слов были сделаны.

Ниже приведен результат поиска (рисунок 4.3)

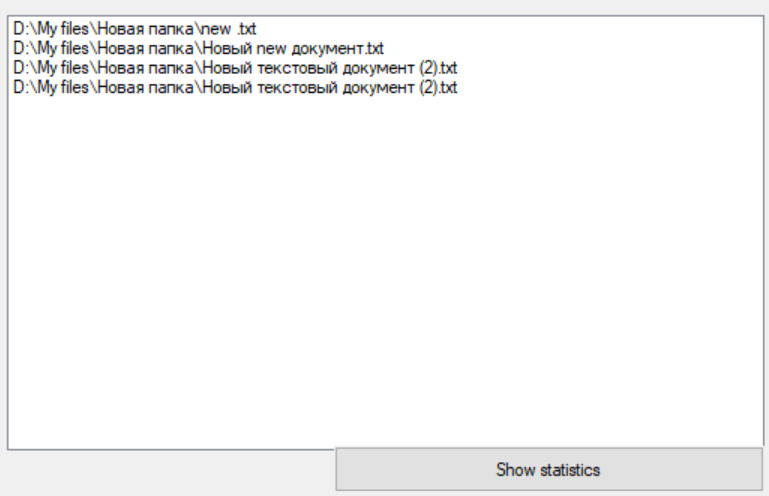


Рисунок 4.3 Результат поиска

Как видно на рисунке 4.3 на ListBox-е данной формы отображаются все полные пути файлов, в которых были найдены искомые слова.

Чтобы просмотреть полную статистику данных слов, воспользуемся кнопкой ShowStatistic и посмотрим результат (рисунок 4.4)

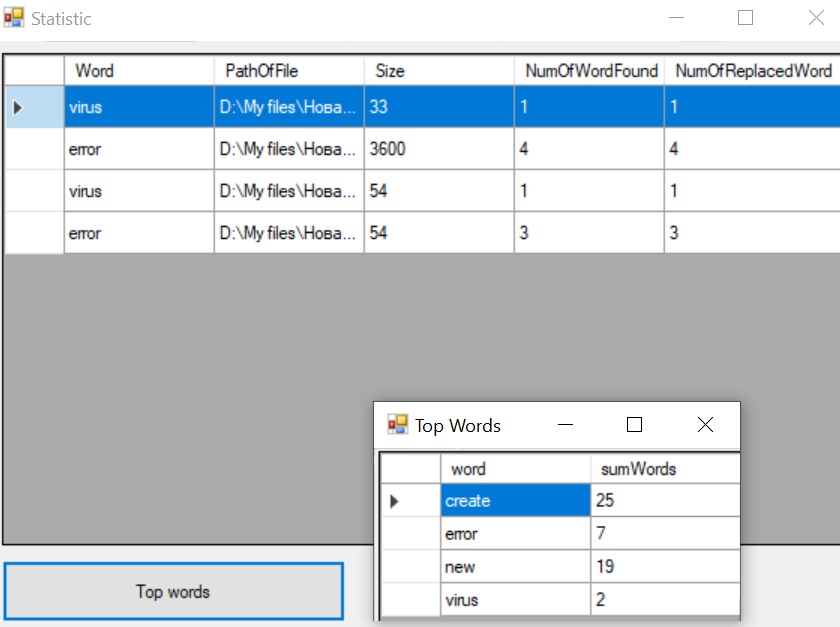


Рисунок 4.4 Найденные файлы и слова в форме Statistic

На данной форме Statistic в таблице можно увидеть всю развернутую статистику данного файла и слова, которое было в нем найдено. Кнопка Top Words выводит топ искомых слов за все время.

Также, данное приложение сохраняет все найденные слова в базу данных (рисунок 4.5)

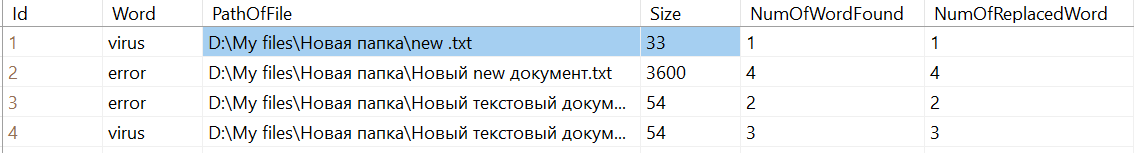


Рисунок 4.5 Найденные файлы и слова записанные в базе данных

Данная статистика сохраняется каждый раз при запуске поиска при условии, что файлы с такими словами найдены.

Также данный найденные файлы (по условию данного задания курсовой работы) копируются отдельно в другую папку в обычной и видоизмененной форме (то есть все искомые слова меняются на \*\*\*\*\*\*\*). Проверим, сохранились ли файлы (рисунок 4.6).

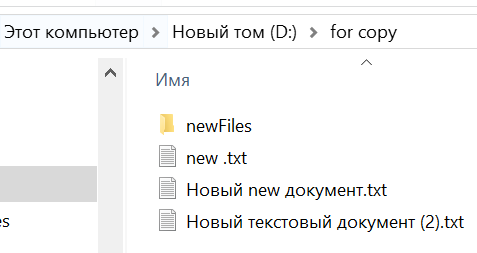


Рисунок 4.6 Скопированные файлы в новой папке

На рисунке 4.6 видно, что данные файлы сохранились в указанный нами путь (D:\for copy) и открыв один из видоизмененных файлов можно увидеть замененные слова (рисунок 4.7)

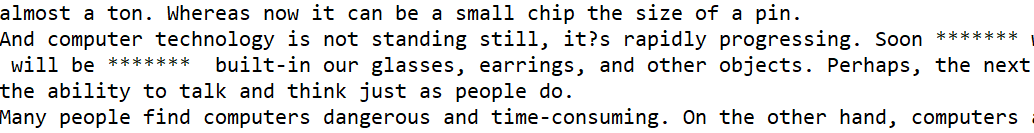


Рисунок 4.7 Открытый файл с замененными словами

Таким образом, отладка и тестирование программы прошло успешно.

Вся программа отработала достаточно быстро (за счёт того, что реализованы операции в разных потоках) и без ошибок.

**5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ И ПРИЛОЖЕНИЕ**

При запуске программы первое, что увидит пользователь, это главную форму поиска Search System (рисунок 5.1).

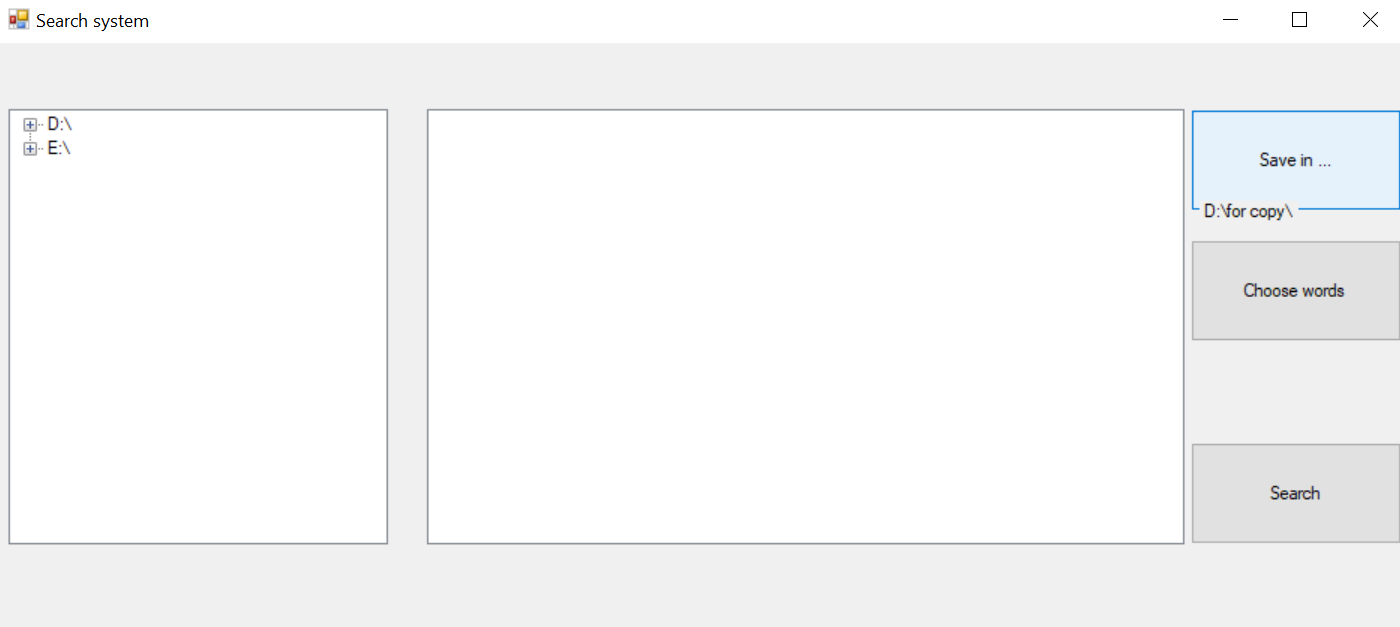


Рисунок 5.1 форма Search System

На этой форме сразу справа можно увидеть все диски, которые есть на данном компьютере и выбрать папку, в которой пользователь хотел бы начать поиск запрещенных слов. Слева на форме изображены три кнопки:

* Save in… - используется для копирования всех найденных файлов в отдельную папку. При наведении курсором на данную кнопку, под ней отображается текущий путь для сохранения. При нажатии на кнопку можно изменить этот текущий путь, выбрав в TreeView другую папку. Чтобы указать функционал пользователю, к этой кнопке был приклеплена подсказка с текстом "Select a folder and click", таким образом делая приложение легкое для понимания.
* Choose Words – кнопка, которая открывает новую форму Words для того, чтобы задать искомые слова.
* и главная кнопка Search, которая отвечает за сам поиск слов.

Также для пользователя есть возможность просмотреть путь, в который будут сохраняться все найденные файлы.

При открытии формы Words, которая отвечает за добавления искомых слов (рисунок 5.2) пользователь увидит поле, в которое можно ввести искомые слова и кнопки:

«+» - добавить

«-» - удалить

«в виде файла» - загрузить слова с файла

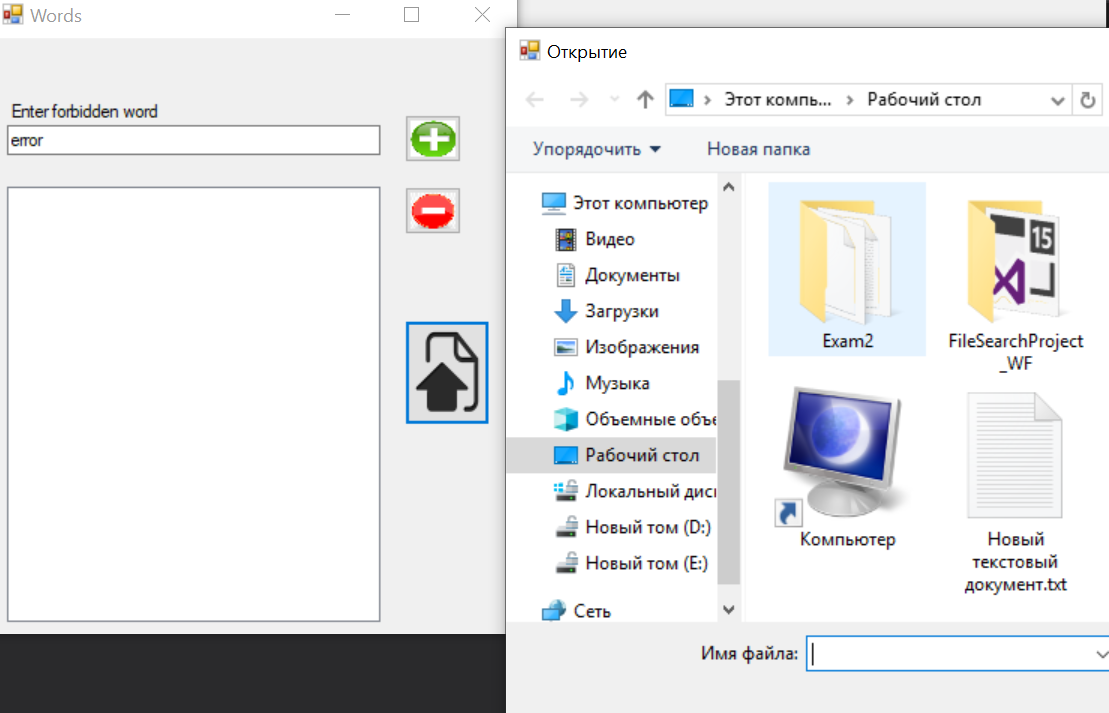


Рисунок 5.2 Форма Words

После добавления слов и указав папку для поиска можно уже нажимать кнопку поиска Search.

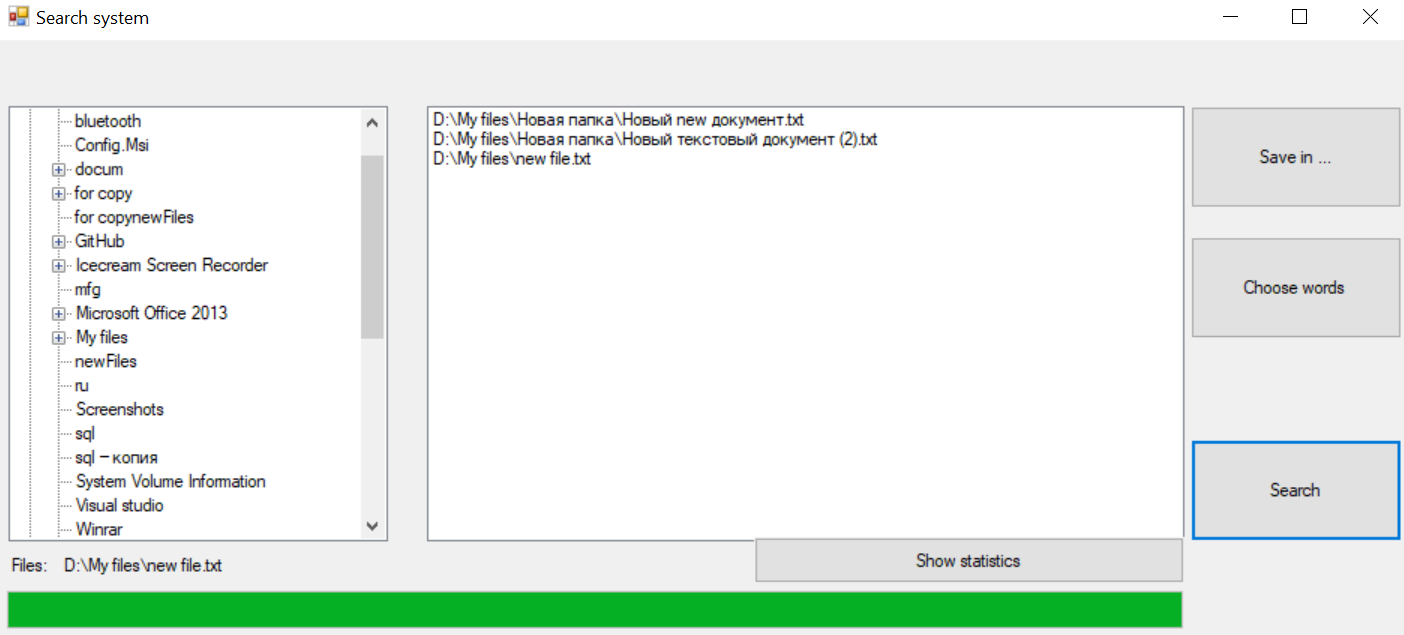


Рисунок 5.3 Результат поиска

После нажатия на кнопку происходит сам поиск, активизируется полоса состояния поиска и указывается путь файлов, в котором на текущий момент идет поиск. Если файлы были найдены с такими искомыми словами, в окне приложения прописываются полные пути файлов, а также сохраняется вся информация в базу данных.

Чтобы просмотреть детальную статистику этих файлов пользователю необходимо нажать кнопку Show Statistics, которая активизируется только при условии, если файлы все же найдены.

После нажатия на кнопку появляется еще одна форма Statistiс (рисунок 5.4). Так же на этой форме можно просмотреть и топ искомых слов за все время, нажав на кнопку Top Words.

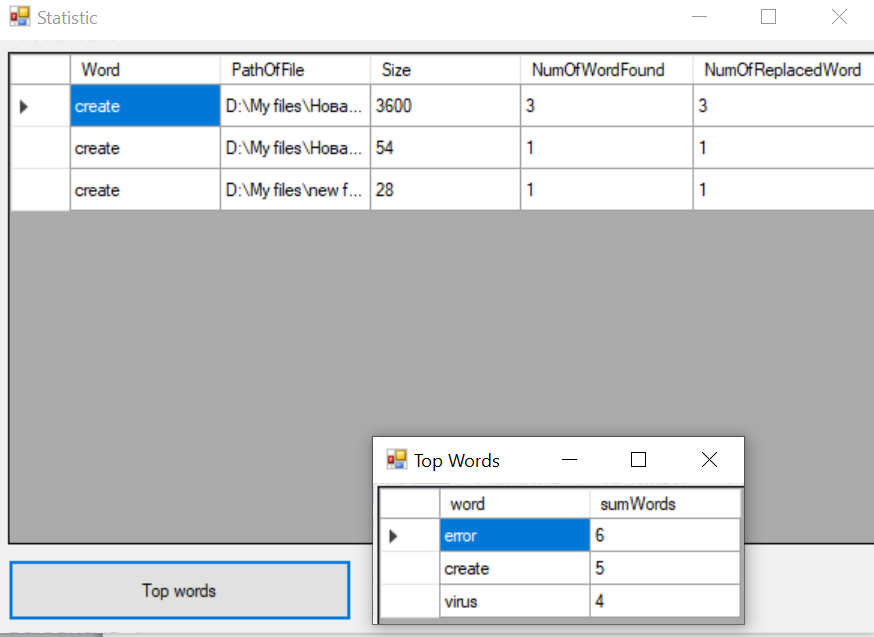


Рисунок 5.4 Форма Statistic и форма Top Words

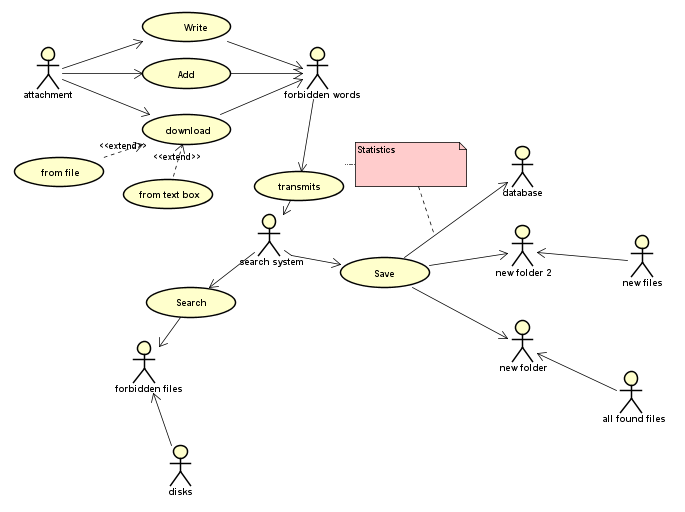
**6 ВЫВОД**

Таким образом, можно сделать вывод, что

* приложение достаточно полезное в применении, так как может решать несколько проблем, таких как поиска файлов на самом компьютере (Google только для локальных папок) с определенными запрещенными словами и поиск вредоносных файлов, которые могут хранить в себе вредоносный код.
* приложение достаточно простое в применении, так как все полезные и нужные кнопки находятся на самой форме и есть подсказки, которые смогут объяснить пользователю не совсем понятные вещи.
* приложение достаточно быстрое и не будет причинять дискомфорт ожиданием выполнения каких-либо действий, так как операции реализованы в разных потоках

**7 ПРИЛОЖЕНИЯ**

Приложение 1. Диаграмма вариантов использования



Приложение 2. Диаграмма последовательности

