

**Пояснительная записка к курсовому проекту**  
по дисциплине:“Технологии программирования”

на тему: “Запись голоса, перевод в текстовый файл без использования интернета”

Выполнил: Проверили:

Черкасов А.Ю. Гужов В.И.

Группа АТ-14

Новосибирск

2023

**Содержание**

[**Введение 3**](#_6sfrhje40yu8)

[**Основная часть 4**](#_8z4qs9x0lpqn)

[Теоретическая часть: 4](#_z0a2dcib5s6f)

[Аналитическая часть: 5](#_yaqguiwj2xga)

[Проектная часть: 8](#_fcw0ide2gnoz)

[Создатель словарей: 8](#_bi41no9cnexv)

[Распознаватель речи: 8](#_j37u51hujc4r)

[**Инструкция по эксплуатации 8**](#_jzc2q4x0qnxb)

[Создатель словарей: 8](#_ggrlqxagcrzv)

[Распознаватель речи: 8](#_2fdhx918glrv)

[**Заключение 8**](#_qbiyfptbfh81)

[**Список источников 8**](#_u5m4h0tt1733)

[**Запуск программы без проекта 8**](#_t9rp6kioqpmn)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Введение

В настоящее время существует множество программ, предназначенных для распознавания речи. Обычно они используются в качестве голосовых ассистентов, переводчиков, поисковиков и других приложений. Однако у большинства из них есть существенный недостаток - для их работы требуется постоянное подключение к интернету. В своей работе я поставил перед собой цель создать аналог таких программ, который не требует подключения к сети. Моя программа должна записывать голосовые сообщения пользователя и преобразовывать их в текст без использования интернета, чтобы обеспечить удобство использования в любой ситуации.

Основные задачи моей программы включают распознавание речи и преобразование ее в текст с использованием локального алгоритма. Она должна работать на операционных системах Windows и обладать функцией автоматического перевода речи в текст при остановке записи. Мой алгоритм должен выполнять все операции локально, без необходимости подключения к интернету. Он должен быть способен обрабатывать различные тембры голоса. Наконец, программа будет написана на языке программирования C#.

# Основная часть

## Теоретическая часть:

Существует множество различных решений от разных производителей для создания программ распознавания речи. Обычно такие программы состоят из двух основных компонентов: платформы распознавания и грамматического словаря.

Люди производят звуки речи, используя различные части своего тела, такие как легкие, ребра, диафрагма, гортань, язык или щеки, чтобы создать поток воздуха и изменить его в голосовом тракте. Часть речи перемещается от одной части к другой, чтобы частично ограничить поток воздуха.

Звуки речи можно классифицировать по степени сужения, начиная с звуков-стопорных согласных (с полной блокировкой потока воздуха), затем фрикативных согласных (с частичной блокировкой и, следовательно, турбулентным потоком воздуха), затем аппроксимирующих согласных (с уменьшенным потоком воздуха без турбулентности) и, наконец, гласных (с полностью свободным потоком воздуха).

Платформа является основным компонентом программы. Она обрабатывает аудиосигнал и, в зависимости от предустановленного набора фонем, ищет соответствие с грамматическим словарем в распознанном тексте. Это необходимо для приведения распознанного текста к знакомому орфографическому виду, так как распознанный текст может сильно отличаться от того, что мы привыкли видеть. Открытых платформ с поддержкой русского языка доступно не так много.

Для достижения поставленных задач нельзя использовать библиотеки распознавания, которые загружают грамматические базы в зависимости от потребностей пользователя. Это означает, что при обработке речи и ее преобразовании в текст нельзя полагаться на интернет для успешного распознавания. Также нет доступных для загрузки открытых речевых словарей для анализа речи, поскольку их большой размер не позволяет широко использовать их.

Необходимо учесть и вопрос производительности. Во-первых, приложение должно распознавать речь асинхронно, поскольку большие словари могут замедлить систему, и если использовать синхронный алгоритм, то производительность будет еще ниже. Поэтому следующий фактор заключается в использовании небольших словарей, специально созданных для конкретных задач.

Также важно учесть удобство пользовательского интерфейса. Программа не должна содержать лишних функций. Программа планируется узкоспециализированной.

## Аналитическая часть:

После анализа всего вышеизложенного, можно сделать вывод, что нам требуется программа, которая сможет считывать внешний файл-словарь, а также отдельная программа для его создания. Рассматривался и вариант основанный на нейросетях, но он был отброшен из-за сложности в освоении и недостатка времени.

Решено было остановится на двух основных вариантах. Первый - использование сторонней библиотеки Microsoft.Speech, а второй - встроенной библиотеки System.Speech в .Net. Отличительной особенностью второго варианта является возможность использования "диктовки". Это означает, что программа не будет ждать завершения фразы, чтобы преобразовать ее в текст. Она также использует предустановленную платформу и грамматический словарь, встроенные в систему Windows. Однако главным недостатком этого варианта является его непригодность для использования, поскольку System.Speech не поддерживает русский язык.

Таким образом, для разработки программы было принято решение использовать библиотеку распознавания речи Microsoft.Speech.SDK. Она обеспечивает работу с .Net и позволяет использовать собственный грамматический словарь. Это отличает ее от конкурирующих решений, которые либо не поддерживают русский язык, либо не могут работать в автономном режиме, либо не имеют версии для разработки на языке C#. Однако у этой библиотеки также есть недостатки. Она представляет собой набор инструментов только для работы с потоковым аудио. Для ее использования требуется установка платформы для распознавания речи Microsoft Speech Platform - Runtime и языкового пакета для нужного языка. Также ее преимуществом является возможность использования XML-файла с грамматиками. Это означает, что при распознавании программа будет использовать логические конструкции для ускорения работы и повышения удобства использования. Однако в нашем случае это нецелесообразно, так как это сильно привязывает программу к готовым решениям. Но этот недостаток не является слишком критическим, поскольку Microsoft.Speech позволяет использовать не только отдельные слова, но и целые фразы для распознавания.

Microsoft Speech Platform SDK 11 предоставляет полный набор средств разработки для создания серверных приложений с поддержкой голосовой связи. Оно позволяет использовать технологии и инструменты Speech Platform SDK 11, чтобы предоставить приложениям возможность распознавать произносимые слова (распознавание речи) и генерировать синтезированную речь (преобразование текста в речь или TTS). Используя этот SDK, можно предоставить пользователям эффективный и естественный способ взаимодействия с приложениями.

Microsoft Speech Platform SDK 11 предлагает базовую

инфраструктуру распознавания речи, которая оцифровывает

акустические сигналы и восстанавливает слова, элементы речи из

аудиовхода, а также обрабатывает сигналы DTMF.

Приложения используют Microsoft.Speech.Пространство имен

Recognition для доступа к этой базовой технологии распознавания

речи и ее расширения путем определения алгоритмов

идентификации определенных фраз или шаблонов слов и действий

с ними, а также путем управления поведением этой

инфраструктуры речи во время выполнения.

Используя классы GrammarBuilder и Choices, вы можете

программно создавать грамматические документы, которые не

зависят от спецификации грамматики распознавания речи 1.0

(SRGS) или не соответствуют ей.

Экземпляры SpeechRecognitionEngine, поставляемые с объектами

Grammar, обеспечивают основной доступ к механизмам

распознавания речи (языкам выполнения сервера), установленным

в системе, и контроль над ними. Среда выполнения Microsoft

Speech Platform Runtime 11 и Speech Platform SDK 11 не включают

никаких движков для распознавания речи на определенном языке.

Необходимо загрузить языковой пакет (движок для распознавания

речи на определенном языке) для каждого языка, на котором вы

хотите выполнить распознавание речи.

Можно использовать SpeechRecognitionEngine для выбора

установленного механизма распознавания речи, запуска,

приостановки или остановки распознавания, загрузки и выгрузки

грамматик, настройки параметров распознавания, настройки ввода

для распознавания и мониторинга входящего аудиосигнала.

Для успешного распознавания используется так называемые

“Телефонные таблицы”. В библиотеке Microsoft.Speech основные

классификации звуков:

Согласные (Microsoft.Speech) - это звуки речи, которые

произносятся при полном (беззвучном) или частичном закрытии

голосового тракта.

Гласные (Microsoft.Speech) - это звуки речи, которые произносятся

открытым голосовым трактом.

Диакритические знаки (Microsoft.Speech) используются для

изменения сегментарных звуков (гласных, согласных, щелчков и

окончаний) с дополнительной фонетической детализацией.

Надсегментарные (Microsoft.Speech) описывают особенности языка

выше уровня отдельных согласных и гласных, такие как тон, длина

и ударение.

Щелчки и произнесения (Microsoft.Speech) - это глухие согласные со

специфическими особенностями произношения.

Тона (Microsoft.Speech) описывают использование высоты тона в

звуках речи для различения лексического или грамматического

значения.

Планируется реализовать создание словаря путем записи данных в файл в формате TXT. Это позволит использовать не только файлы, созданные с помощью нашей программы, но и другие сторонние файлы для заполнения словаря.

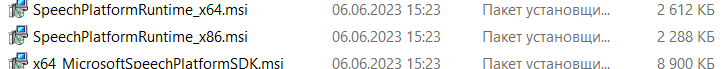
## Проектная часть:

### Создатель словарей:

### Программа предоставляет две возможности, создать новый словарь или открыть уже существующий по нажатию соответствующих кнопок. В лейбл выводится информация об успешности открытия,создания файла или добавления слова в словарь. По нажатию кнопки “Добавить слово” слово написанное в TextBox добавляется в словарь.

### Распознаватель речи:

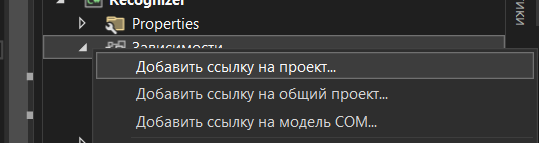
После установки SDK MicrosoftSpeechPlatform(3 файл), Runtime(1 и 2 файлы) и аддона для русского языка(4 файл). Мы можем начинать создание программы.



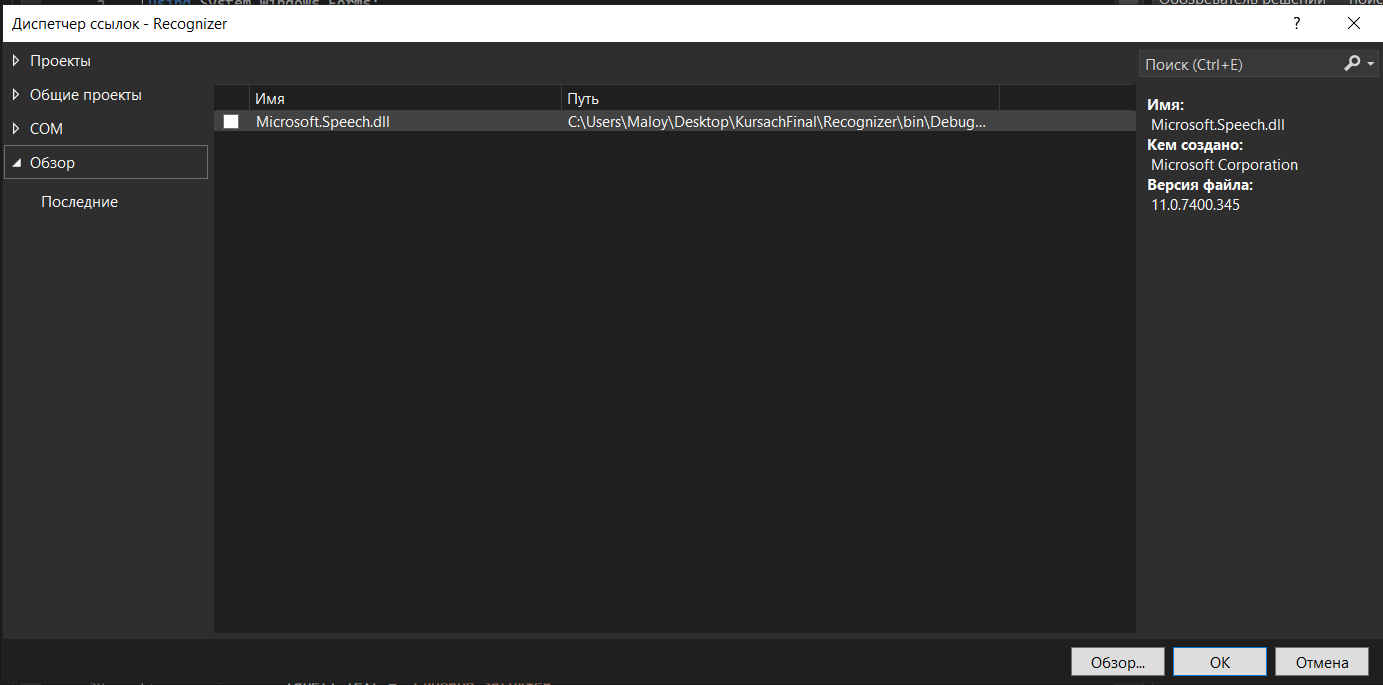


Для работы нашей программы необходимо добавить стороннюю библиотеку Microsoft.Speech.dll. Добавляем её следующим образом:

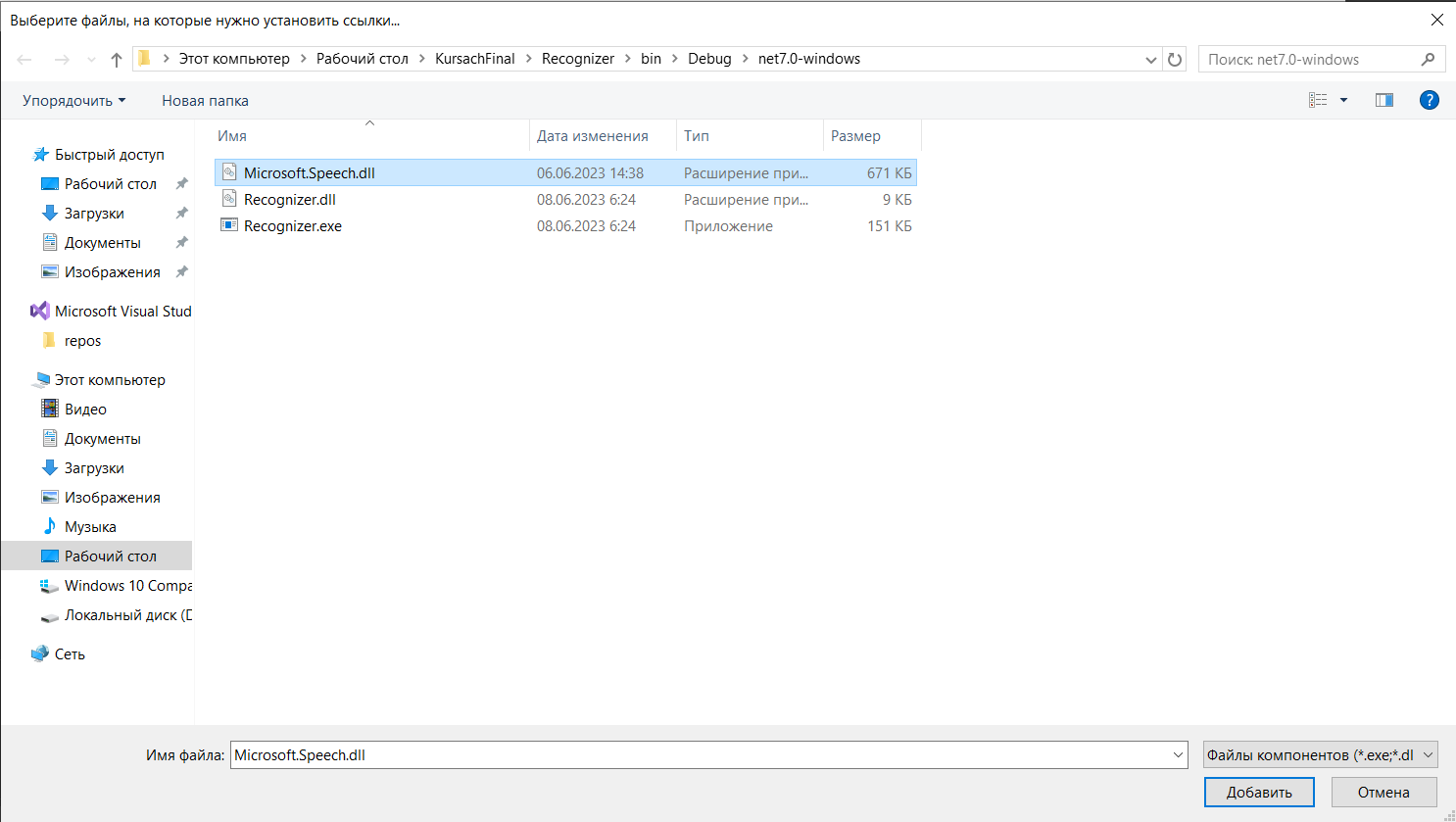
1. Нажимаем ПКМ по “Зависимости” и выбираем “добавить ссылку на проект”



2. В открывшимся меню нажимаем “обзор”



3. Ищем файл MicrosoftSpeech.dll он находится там, куда его установила MicrosoftSpeechPlatformSDK. Для удобства я перекинул его в файлы проекта.



4. После нажимаем добавить и можем начинать разработку программы.

Сначала выводится диалоговое окно, в котором надо выбрать txt файл со словарем, в котором хранятся слова, которые сможет распознавать наша программа. Словарь создается сторонней программой Создатель словарей и содержит по одному слову или словосочетанию на каждой строке. Далее создается новый элемент из библиотеки MicrosoftSpeech

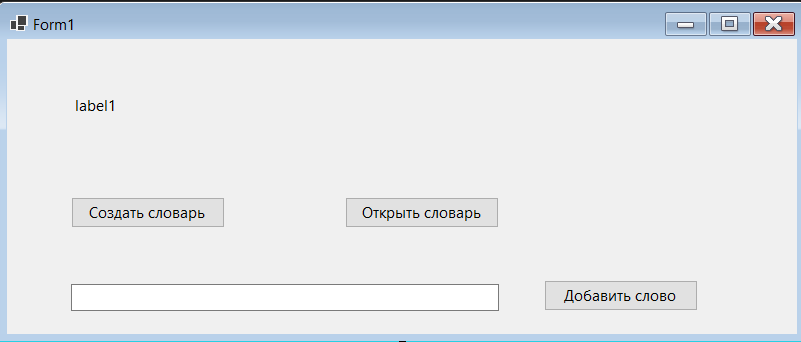
SpeechRecognitionEngine. Чтобы его использовать нужно объявить и заполнить несколько переменных. Создаем набор слов для распознавания Choices передавая в него строки из считанного вначале файла. Создаем GrammarBuilder в который передаем набор слов и информацию о языке,на котором они написаны CultureInfo. После чего создаем словарь Grammar передавая в конструктор GrammarBuilder. Затем загружаем словарь в engine, устанавливаем микрофон по умолчанию для записи звука, передаем функцию распознавания speechRecognizer, которая при совпадении больше 0.6(из 1) записывает слово в textBox. Затем включаем мод Multiple распознавания, чтобы программа распознавала не первое распознанное слово, а дослушала до конца(например между словом “как” и словосочетанием “как дела” есть разница, но Single распознавание в обоих случаях распознает “как”)

Можно сохранить результат распознавания по нажатию кнопки ”Сохранить расшифровку”. Приложение позволяет выбрать папку и имя файла.

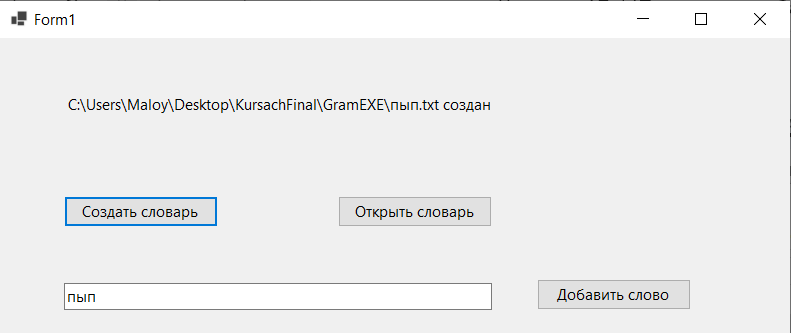
# Инструкция по эксплуатации

## Создатель словарей:

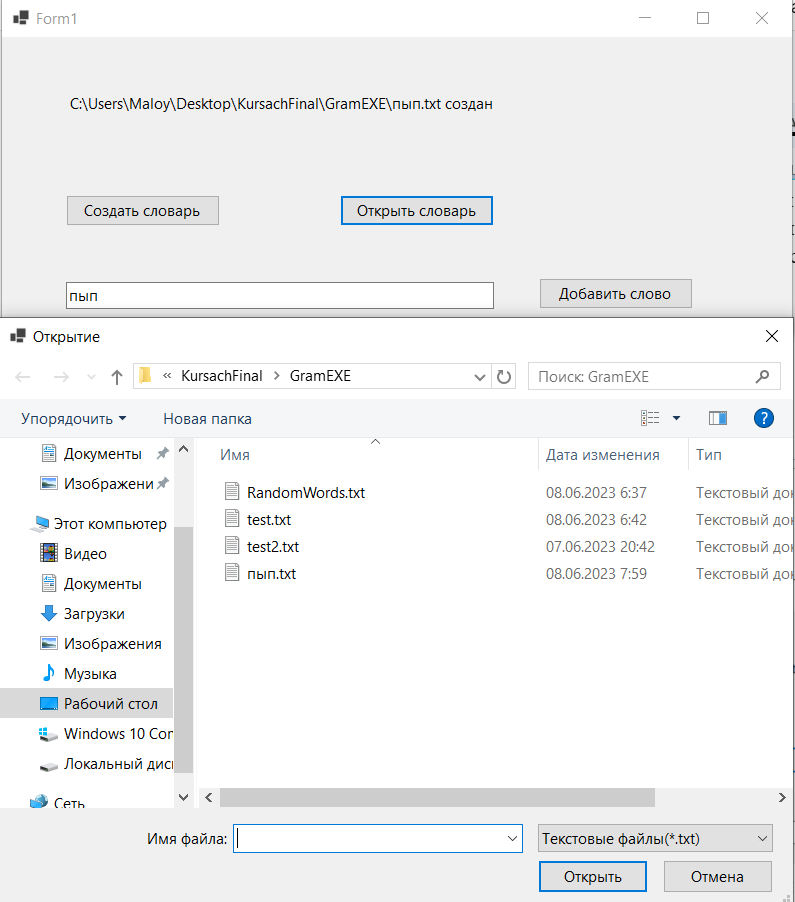
Программа реализовывает добавление новых слов в словарь, существующий или созданный этой программой.

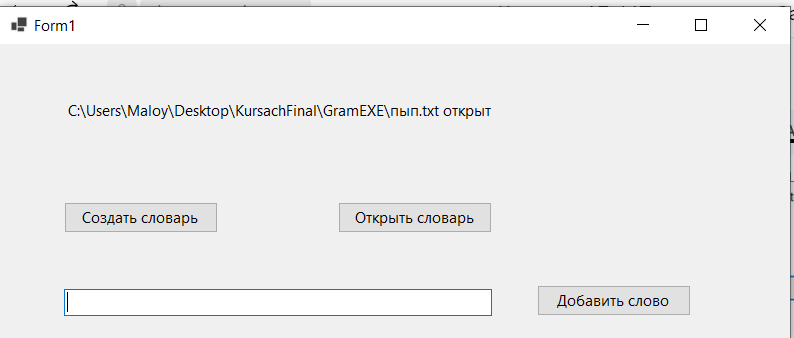


По нажатию кнопки “создать словарь” в месте расположения exe txt файл с названием написанным в TextBox(оно должно быть длинной хотя бы 1 символ, чтобы файл создался).

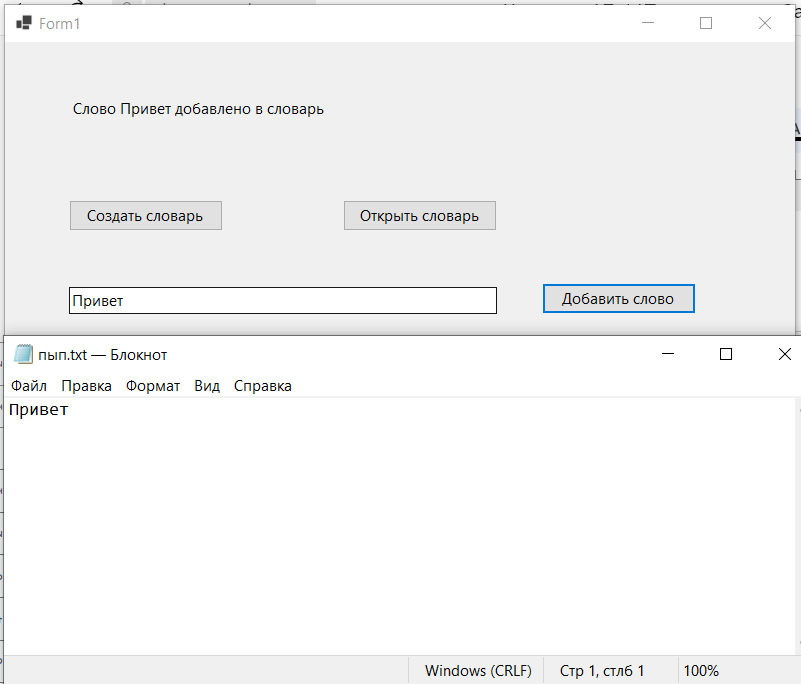


По нажатию кнопки “Открыть словарь” открывается меню, в котором можно выбрать txt файл который надо открыть.



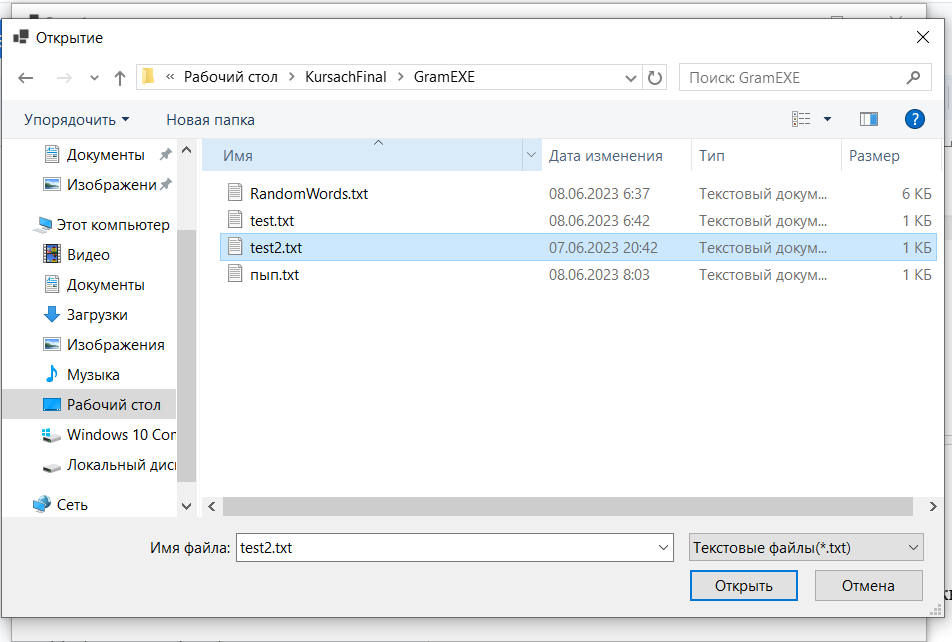


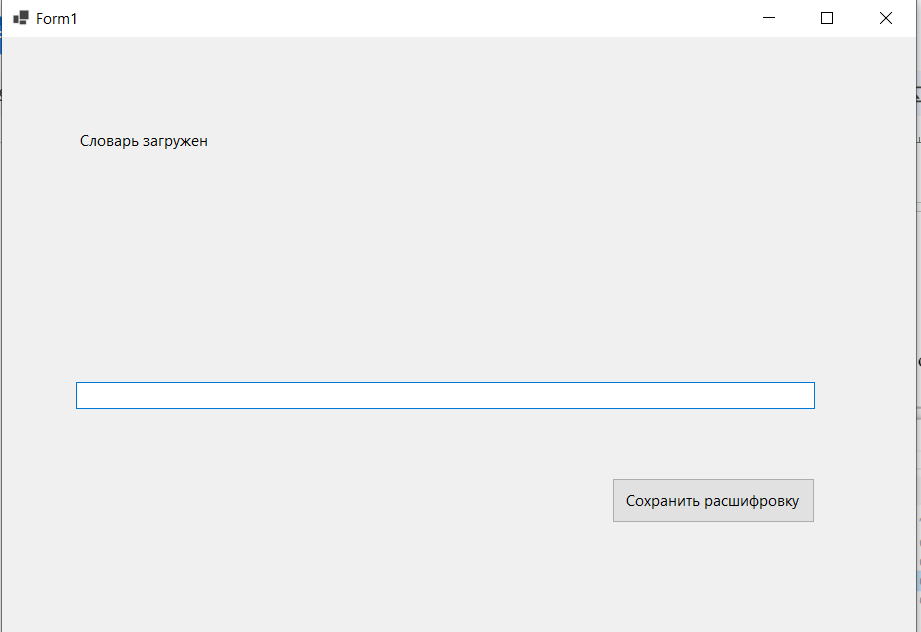
По нажатию кнопки добавить слово в выбранный словарь добавляется слово написанное с новой строки.



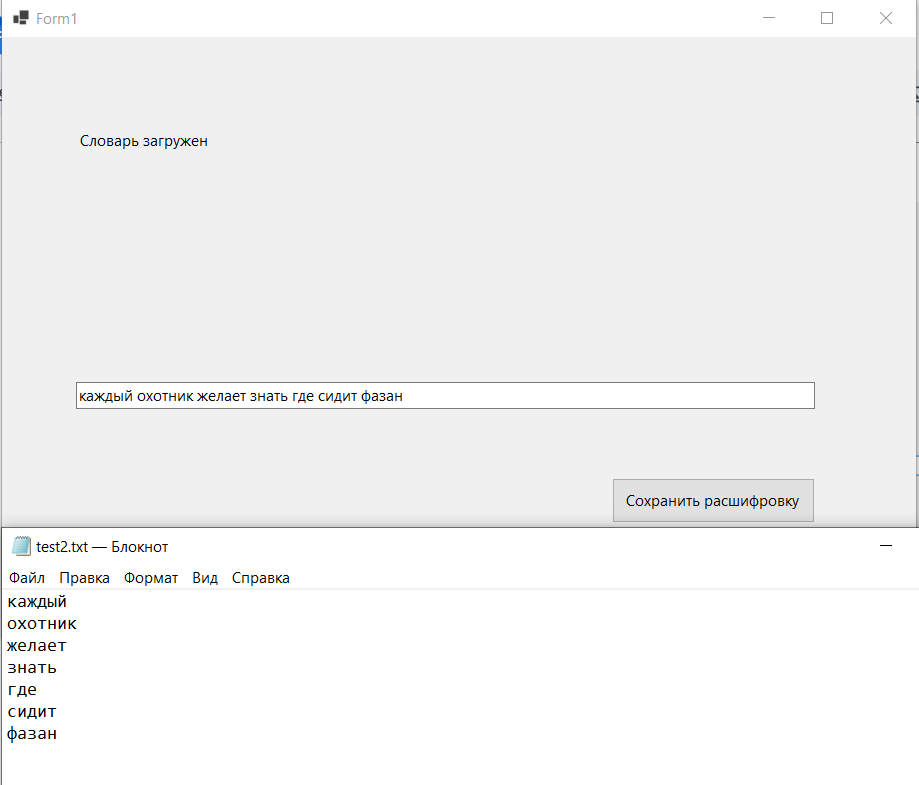
## Распознаватель речи:

По открытию программы появляется окно выбора, где нужно выбрать файл со словарем.



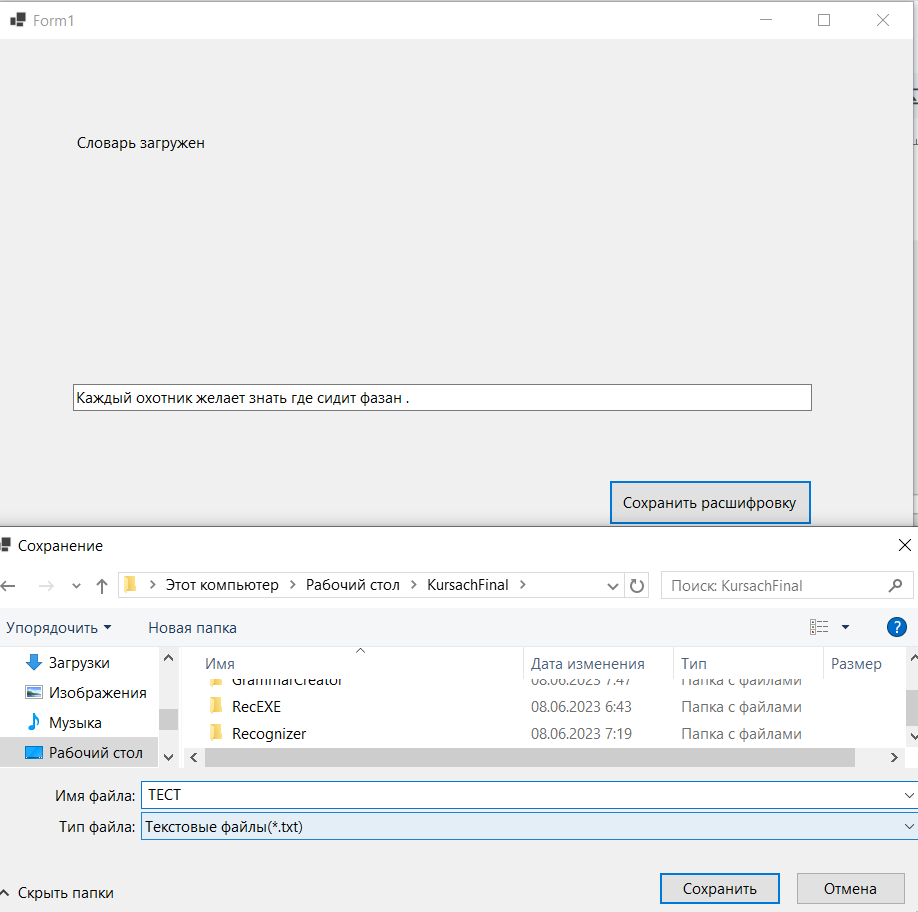
Если словарь открылся успешно, то появляется надпись “Словарь загружен”

После этого можно диктовать фразы состоящие из слов в словаре.

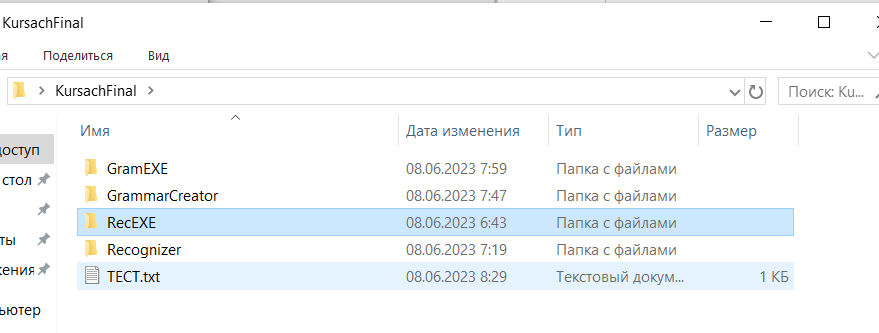


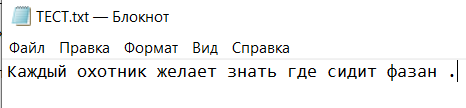
Можно отредактировать расшифровку, а затем нажать кнопку “Сохранить расшифровку”.

Откроется окно с выбором места сохранения файла и его имени.



После нажатия кнопки “Сохранить” в выбранном месте будет создан txt файл с расшифровкой.





# Заключение

В результате работы над курсовым проектом были созданы две программы. Первая для создания словарей слов, которые будут распознаваться. Вторая для распознавания фраз, состоящей из слов находящихся в загруженном словаре. Полученные программы способны запускаться без проекта и позволяют сохранять результат.

# Список источников

# 

<https://documentation.help/Microsoft-Speech-Platform-SDK-11/> - документация Microsoft Speech.

<https://www.youtube.com/watch?v=-qpVafFnthU> - видео-туториал по созданию программы способной распознавать речь на основе Microsoft Speech.

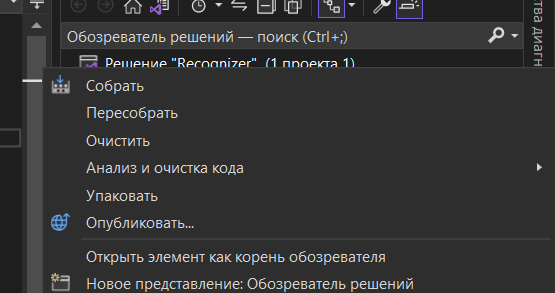
“Технология разработки объектно-ориентированных программ на

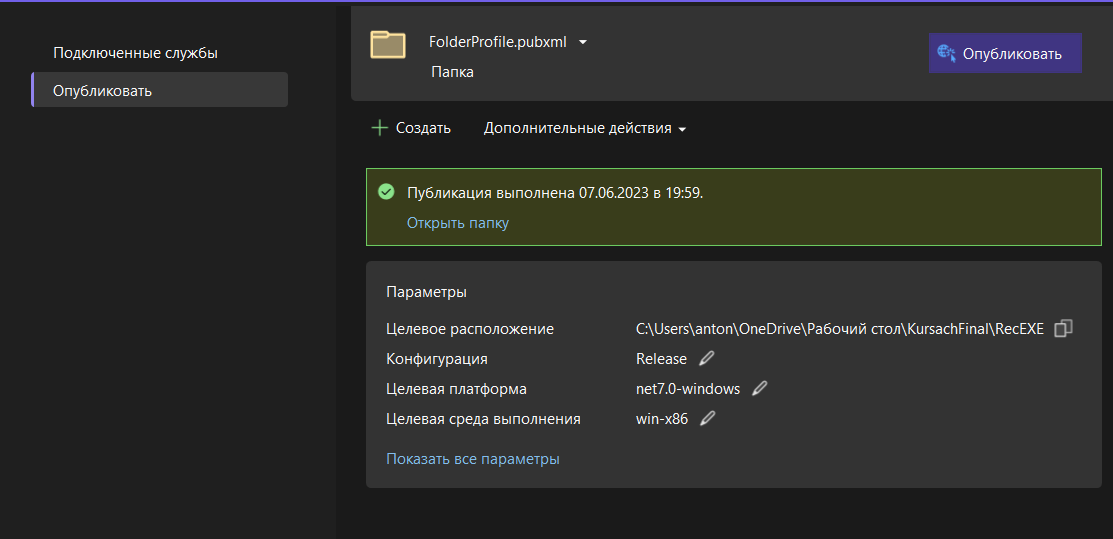
языке C# в среде Visual Studio.Net.” Васюткина И.А

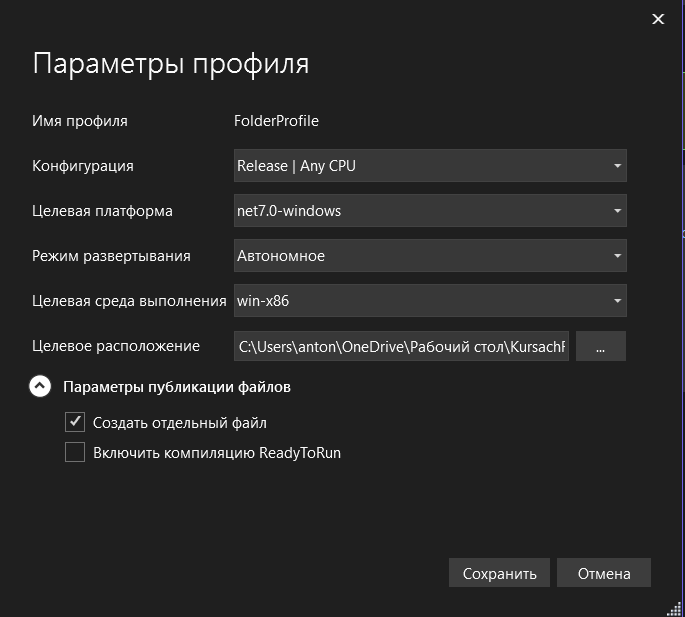
<https://habr.com/ru/sandbox/103574/> - статья о Microsoft Speech.

# Запуск программы без проекта

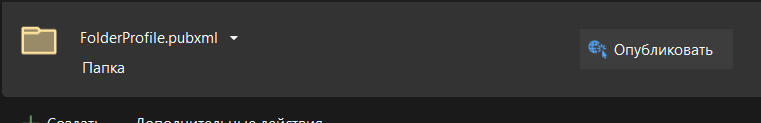
В проекте в Visual Studio нажимаем на   
“Опубликовать”.



В открытом меню надо выбрать “Показать все параметры”

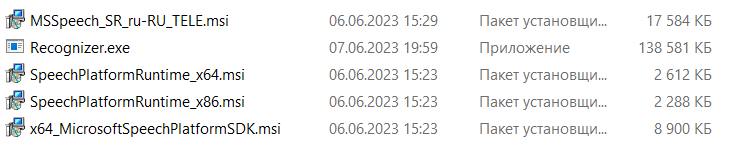
Там выбрать “Режим развертывания - Автономное” и поставить галочку напротив “Создать отдельный файл”. Также выбрать папку в которой будет располагаться exe файл.

Затем нажать “Опубликовать”.



После определенного времени exe файл появится в выбранной папке. Вместе с ним появятся несколько других файлов, но их можно удалить, exe будет работать без них.

# Инструкция по установке

Что бы Recognizer.exe начал работать, нужно установить 4 файла из RecEXE.

Сначала устанавливается x64\_MicrosoftSpeechPlatformSDK, затем SpeechPlatformRuntime\_x64 и SpeechPlatformRuntime\_x86. Самым последним MSSpeach\_SR\_ru-RU\_TELE. После этого программа будет работать.