Windows 10

Лабораторный практикум

Запуск приложений при помощи голосовых команд

Октябрь 2015 года

Microsoft

Обзор

Приложения Windows 10 могут использовать интерфейс Cortana, чтобы взаимодействовать с пользователями через удобные и настраиваемые голосовые команды. Cortana может запустить ваше приложение на переднем плане, или взаимодействовать с данными приложения в фоновом режиме.

Специфические для приложения голосовые команды начинаются с префикса, как правило, с имени приложения или ключевого слова, что позволяет устранить неоднозначность в отношении других приложений, которые могут иметь аналогичные команды. Вы можете определить опции произнесения имени приложения перед или после команды и выбрать схему поведения после запуска приложения.

В этой работе вы создадите файл определения голосовых команд и будете добавлять команды, чтобы запустить своё приложение и работать с ним.

Цели

Эта лабораторная работа покажет вам как:

- Создать файл определения голосовых команд
- Запустить своё приложение, используя голосовую команду
- Использовать переключение для выбора входящих голосовых команд
- Передавать информацию из голосовой команды в приложение
- Переключать вид приложения по входящей голосовой команде
- Запускать фоновую задачу из голосовой команды
- Возвращать письменные и голосовые ответы Cortana из своего приложения

Системные требования

Чтобы выполнить этот практикум, необходимо обладать следующим набором программных инструментов:

- Microsoft Windows 10 настраивается на один из языков и регионов, в которых поддерживается Cortana
    - В сентябре 2015 года Cortana стала доступной в следующих странах/регионах: Китай, Франция, Германия, Италия, Испания, Соединённое Королевство и Соединённые Штаты. Cortana доступна на следующих языках: Китайский (упрощённый), английский (Великобритания), английский (США), французский, итальянский, немецкий и испанский языки
    - Для того, чтобы пользоваться Cortana, необходимо произвести все настройки на один и тот же язык:
        - Языки (язык вашего устройства)
        - Разговорный язык (должен быть установлен языковой пакет)
        - Страна или регион
- Microsoft Visual Studio 2015

Настройка

Вы должны осуществить следующие шаги для подготовки своего компьютера для этого курса:

1. Установите Microsoft Windows 10.
2. Установите Microsoft Visual Studio 2015.

Упражнения

Этот лабораторный практикум включает следующие упражнения:

1. Запуск при помощи голосовых команд
2. Передача голосового параметра в приложение
3. Ответ на голосовую команду при помощи фоновой задачи

Расчётное время для завершения работы: **От 30 до 45 минут**.

Упражнение 1: Запуск при помощи голосовых команд

Голосовые команды представляют удобную альтернативу ручному запуску вашего приложения. Для использования голосовых команд вам нужно создать файл голосового описания, указывающий команду запуска вашего приложения и соответствующий ответ Cortana. Зарегистрируйте команды с Cortana и используйте точку старта OnActivated для старта приложения.

Задача 1 – Создать шаблон приложения Universal Windows

Начнём с создания проекта из шаблона пустого приложения.

1. В новой версии Visual Studio 2015 выберите **File (Файл) -> New (Новый) -> Project (Проект)**, чтобы открыть диалоговое окно New Project (Новый проект). Далее **Installed (Установленное) > Templates (Шаблоны) > Visual C# > Windows > Universal**, а затем выберите шаблон **Blank App** приложения (Universal Windows).

2. Назовите свой проект **SpeechRecognition** и выберите местоположение файловой системы, в которой вы сохраните результаты прохождения практикума. На диске **C** создана папка под именем **"HOL"**, информация о которой будет представлена в скриншотах во время прохождения всех практикумов.

   Не изменяйте настройки, установленные для **Create new solution (Создания нового решения)** и **Create directory for solution (Создания папки для решения)**. Вы можете снять галочки как с **Add to source control (Добавить в систему контроля версий)**, так и с **Show telemetry in the Windows Dev Center (Отобразить телеметрию в Windows Dev Center)**, если не хотите использовать эти возможности. Нажмите **OK** для создания проекта.

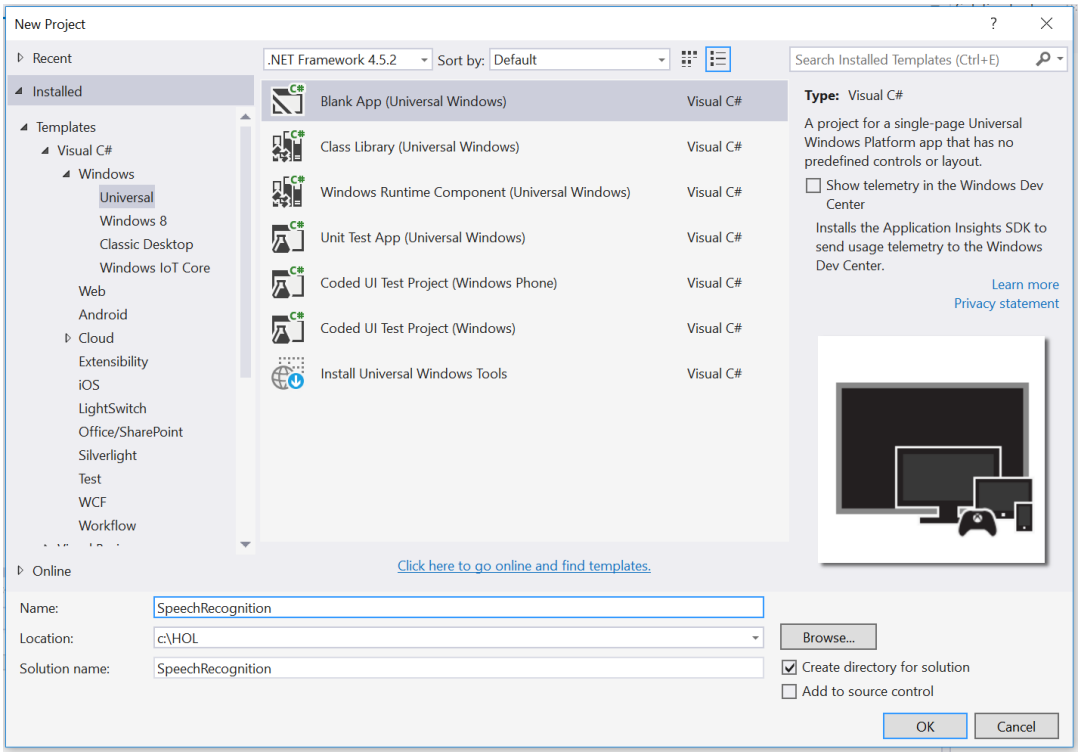

Рисунок 1

Создание нового проекта приложения в Visual Studio 2015.

3. Настройте Текущую конфигурацию решения на **Отладку** и Платформу решений в **x86**. Выберите **Локальный компьютер** из выпадающего меню Цели отладки.

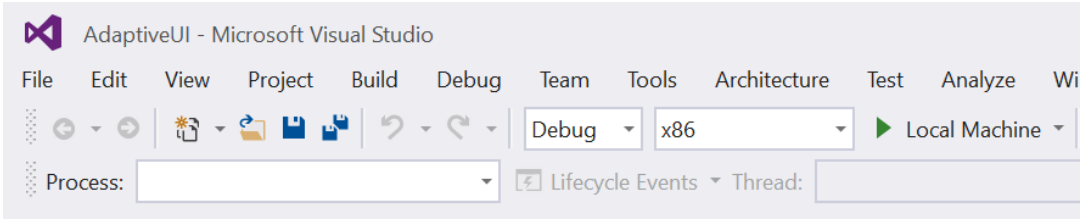

Рисунок 2

Сконфигурируйте свое приложение таким образом, чтобы оно запускалось на Локальном компьютере.

4. Скомпилируйте и запустите своё приложение. Вы увидите окно шаблона приложения со счетчиком частоты кадров, активированном по умолчанию для отладки.

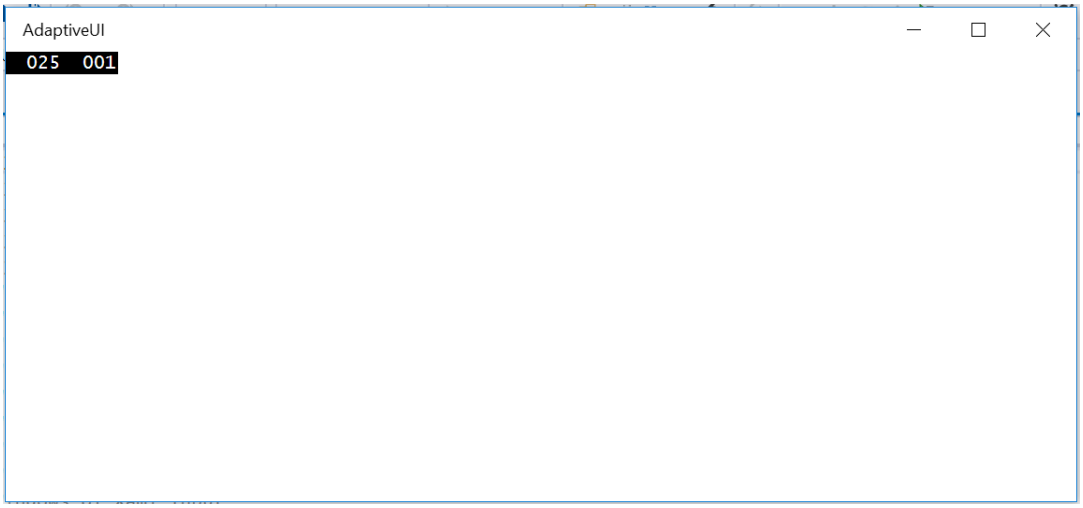

Рисунок 3

Пустое универсальное приложение, запущенное в режиме настольного компьютера.

Примечание: Счетчик частоты кадров является инструментом, используемым в процессе отладки, который помогает следить за производительностью вашего приложения. Он полезен для тех приложений, которые требуют интенсивной графической обработки, однако не удобен для простых приложений, которые будут создаваться вами в данном практическом курсе.

В шаблоне пустого приложения директива препроцессора для активации или отключения счетчика частоты кадров находится на **App.xaml.cs**. Счетчик частоты кадров может перекрывать или скрывать контент вашего приложения, если не убрать его. При выполнении данных работ вы можете отключить его, установив значение **DebugSettings.EnableFrameRateCounter** в **false**.

5. Вернитесь к Visual Studio и остановите отладку.

Задача 2 – Создайте файл определения голосовых команд

Схема голосовой команды определяется в XML-файле. В этой задаче вы создадите простую схему для голосового запуска приложения.

1. Щёлкните правой кнопкой мыши по наименованию проекта и выберите **Add (Добавить) > New Item** (Новый элемент).

2. Выберите тип XML-файлов и присвойте ему имя **VoiceCommands.xml**.

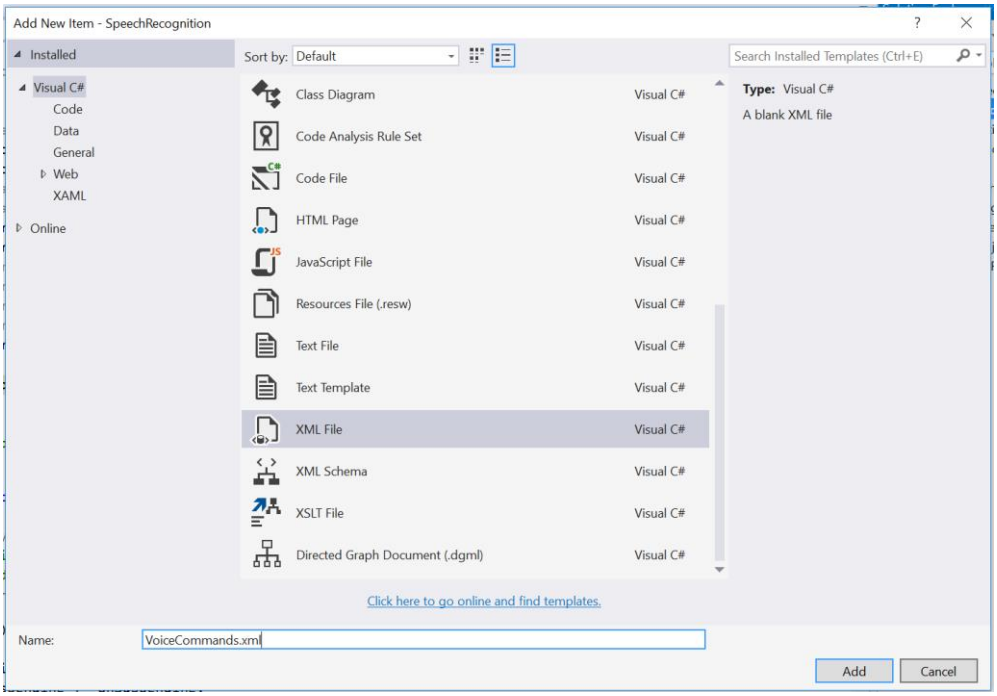

Рисунок 4

Создайте XML файл голосовых команд.

3. Откройте VoiceCommands.xml. После заголовка версии XML-файла добавьте элемент VoiceCommands и атрибут xmlns, ссылающийся на формат схемы голосовых команд . Добавьте элемент, описывающий набор команд, для языка EN-US (или другого используемого вами в работе поддерживаемого языка).

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<VoiceCommands xmlns="http://schemas.microsoft.com/voicecommands/1.2">
    <CommandSet xml:lang="en-us" Name="HoLCommandSet_en-us">
    </CommandSet>
    <!—Опционально: второй набор команд для другого языка -->
    <!-- <CommandSet xml:lang="de-de" Name="HoLCommandSet_de-de"/> -->
</VoiceCommands>
```

Примечание: Мы добавили язык **EN-US** для этого примера, но вы можете добавить свой собственный набор команд на другом языке. Например, языковый тег для Германии был бы xml:lang="de-de". Список регионов и языков, которые поддерживает Cortana, находится на http://windows.microsoft.com/en-us/windows-10/cortanas-regions-and-languages

Если вы решаете добавить другой набор команд на поддерживаемом языке, убедитесь в том, что каждый раз добавляя одну из EN-US команд, вы одновременно добавляете эквивалентную команду на этом языке.

4. Добавьте голосовой префикс для вашего приложения. Префикс– это слово или фраза, которую ваши пользователи должны сказать, чтобы указать ваше приложение в качестве получателя команд.

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<VoiceCommands xmlns="http://schemas.microsoft.com/voicecommands/1.2">
  <CommandSet xml:lang="en-us" Name="HoLCommandSet_en-us">
    <CommandPrefix> Hands on Labs, </CommandPrefix>
</CommandSet>
```

Примечание: Запятая после префикса команды необязательна. Если вы решите добавить её, она будет означать небольшую паузу между префиксом команды и самой командой. При распознавании команды Cortana также кратко отобразит префикс и команду именно так, как они записаны здесь.

Если вы решите добавить дополнительный набор голосовых команд на другом языке, добавьте префикс команды на этом языке в соответствующий набор команд.

5. Наша цель - запуск приложения с помощью голосовой команды. Создайте команду запуска с помощью элемента **ListenFor (Прослушать)**, который определяет слово или слова, которые будут распознаны для этой команды, с помощью элемента **Feedback (Обратная связь)**, который определяет текст подтверждения, проговариваемый Кортаной после пользователя при распознавании команды, и элемента **Navigate (Переход)**. Добавьте элемент **Example (Пример)** для новой команды к содержащему её **CommandSet (Набору команд)**.

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<VoiceCommands xmlns="http://schemas.microsoft.com/voicecommands/1.2">
  <CommandSet xml:lang="en-us" Name="HoLCommandSet_en-us">
    <CommandPrefix> Hands on Labs, </CommandPrefix>
    <Example> Launch </Example>

    <Command Name="LaunchApp">
      <Example>launch</Example>
      <ListenFor>launch</ListenFor>
      <Feedback>Opening your speech recognition app</Feedback>
      <Navigate />
    </Command>
</CommandSet>
</VoiceCommands>
```

Примечание: Элемент <Navigate/> означает, что в ответ на команду произойдет запуск приложения на переднем плане. Альтернативой запуска на переднем плане является определение компонента WinRT, который обрабатывает скрытое взаимодействие с данными приложения. Вы можете узнать больше об определениях голосовых команд по ссылке: https://msdn.microsoft.com/en-us/library/windows/apps/dn722331.aspx

Задача 3 – Зарегистрируйте определения голосовых команд

Необходимо зарегистрировать определения голосовых команд (VCD) в переопределении функции **OnLaunched**. Не существует простого способа проверить, был ли ранее импортирован файл VCD, поэтому удобно каждый раз при запуске приложения регистрировать последнюю версию.

1. Откройте **App.xaml.cs**, добавьте ключевое слово **async** в переопределение **OnLaunched** и добавьте следующие строки:

```
C#
protected override async void OnLaunched(LaunchActivatedEventArgs e)
{

    //#if DEBUG
    //            if (System.Diagnostics.Debugger.IsAttached)
    //            {
    //                this.DebugSettings.EnableFrameRateCounter = true;
    //            }
    //#endif

    var storageFile = await
Windows.Storage.StorageFile.GetFileFromApplicationUriAsync(new Uri("ms-
appx:///VoiceCommands.xml"));

    await
Windows.ApplicationModel.VoiceCommands.VoiceCommandDefinitionManager.Instal
lCommandDefinitionsFromStorageFileAsync(storageFile);

    Frame rootFrame = Window.Current.Content as Frame;
```

Примечание: Определение голосовых команд, (VCD) будет установлено при первом запуске приложения. Необходимо первый раз открыть своё приложение из меню запуска для того, чтобы убедиться, что голосовые команды зарегистрированы.

Задача 4 – Активация по голосовой команде

Если ваше приложение запускается через голосовую команду, то его тип запуска будет **Activated (активировано)**. Соответственно, мы будем обрабатывать входящие голосовые команды в переопределении функции **OnActivated**. Необходимо при запуске проверить, было ли приложение запущено через голосовую команду или нет, и соответствующим образом скорректировать поведение приложения.

1. Создайте переопределение **OnActivated** в **App.xaml.cs**. Запуск голосовой команды соответствует вызову **OnActivated**, так что вы уже здесь сможете обработать входящую голосовую команду.

C#

```
protected override void OnActivated(IActivatedEventArgs args)
{
    base.OnActivated(args);
}
```

2. Добавьте команду для обработки случая **ActivationKind.VoiceCommand**, и вызовите метод **HandleVoiceCommand**. Метод **HandleVoiceCommand** вы создадите на следующем этапе.

C#

```
protected override void OnActivated(IActivatedEventArgs args)
{
    switch (args.Kind)
    {
        case ActivationKind.VoiceCommand:
            HandleVoiceCommand(args);
            break;

        default:
            break;
    }
    base.OnActivated(args);
}
```

3. Добавьте пространство имен **System.Diagnostics к App.xaml.cs**.

C#

```
using System.Diagnostics;
```

4. Создайте метод **HandleVoiceCommand**. Этот метод обрабатывает входящую голосовую команду. Вы задали команду **LaunchApp (запуск приложения)** в своём файле определений голосовых команд в разделе 2.

C#

```
private void HandleVoiceCommand(IActivatedEventArgs args)
{
    var commandArgs = args as VoiceCommandActivatedEventArgs;
    var speechRecognitionResult = commandArgs.Result;
    var command = speechRecognitionResult.Text;

    var voiceCommandName = speechRecognitionResult.RulePath[0];
    var textSpoken = speechRecognitionResult.Text;

    Debug.WriteLine("Command: " + command);
    Debug.WriteLine("Text spoken: " + textSpoken);

    switch (voiceCommandName)
    {
        case "LaunchApp":
            break;
```

```
            default:
                break;
        }
    }
```

Примечание: Промежуточная печать команды через Debug.WriteLine будет полезна позднее при отладке кода. Вы можете добавить дополнительную печать, если хотите увидеть значения commandArgs и speechRecognitionResult. Далее в этой задаче мы рассмотрим некоторые приемы отладки приложений, запускаемых через голосовые команды.

5. Для успешного запуска приложения и перехода на страницу, нам нужно будет воспроизвести процедуру запуска приложения в функции OnLaunched. Шаблон приложения Visual Studio создаёт root frame и активирует окно, но в функции OnActivated эти действия не производятся. Добавьте соответствующий код для отображения окна в функцию OnActivated.

C#

```
protected override void OnActivated(IActivatedEventArgs args)
{
    Frame rootFrame = Window.Current.Content as Frame;

    if (rootFrame == null)
    {
        rootFrame = new Frame();
        rootFrame.NavigationFailed += OnNavigationFailed;
        Window.Current.Content = rootFrame;
    }

    switch (args.Kind)
    {
        case ActivationKind.VoiceCommand:
        HandleVoiceCommand(args);
        break;

        default:
        break;
    }

    Window.Current.Activate();

    base.OnActivated(args);
}
```

Примечание: Чтобы избежать дублирования в реальном приложении, обычно имеет смысл создать общий код запуска в отдельной функции, который выполнится для запущенных и активированных приложений. Template10 демонстрирует более унифицированный способ обработки этих важных действий при запуске приложения. Для ознакомления с более подробной информацией о Template10, посетите страницу https://github.com/Windows-XAML/Template10

6. Передайте ссылку на rootFrame методу **HandleVoiceCommand** в дополнение к **args**. Вам понадобится контекст корневого кадра для перехода на страницу.

C#

```
switch (args.Kind)
    {
        регистр ActivationKind.VoiceCommand (тип активации. Голосовая
команда):
        HandleVoiceCommand (аргументы, rootFrame);
        разрыв;

        по умолчанию:
        разрыв;
    }
```

7. В методе **HandleVoiceCommand** добавьте параметр **frame** и используйте его для перехода на страницу **MainPage**:

C#

```
private void HandleVoiceCommand(IActivatedEventArgs args, Frame frame)
{
    var commandArgs = args as VoiceCommandActivatedEventArgs;
    var speechRecognitionResult = commandArgs.Result;
    var command = speechRecognitionResult.Text;
    var voiceCommandName = speechRecognitionResult.RulePath[0];
    var textSpoken = speechRecognitionResult.Text;
    switch (voiceCommandName)
    {
        case "LaunchApp":
            frame.Navigate(typeof(MainPage));
            break;
        default: break;
    }}
```

8. Откройте MainPage.xaml и добавьте заголовок страницы. Текст поможет определить, что переход был совершен в момент запуска приложения.

XAML

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
    <TextBlock Text="Speech Recognition and Voice Commands"
FontWeight="Light" FontSize="20" Margin="12" />
</Grid>
```

9. Создайте и запустите своё приложение на локальном компьютере.

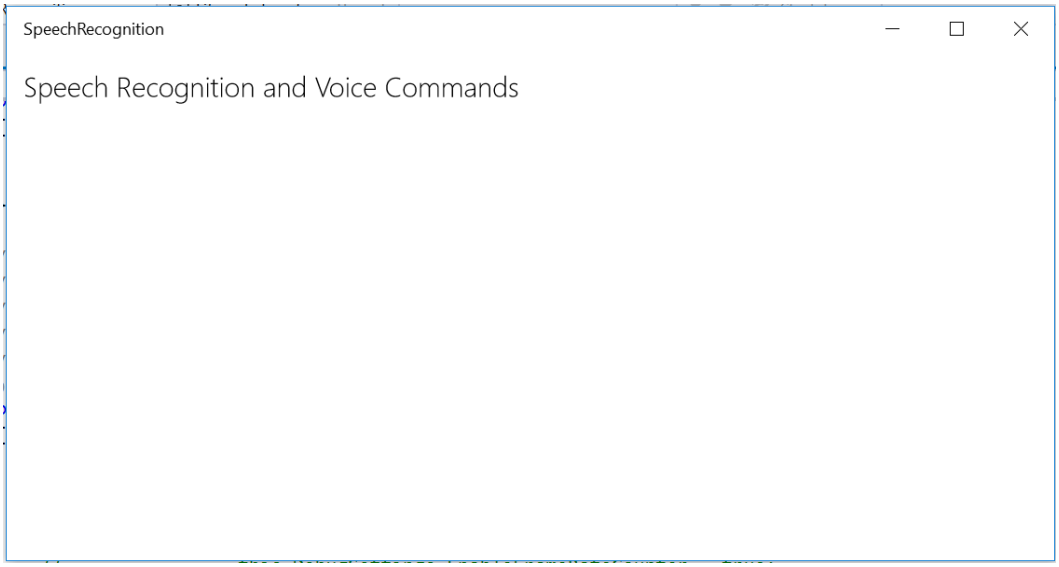

Рисунок 5

Приложение, выполняющееся на Локальном компьютере.

10. Закройте приложение. Откройте свойства проекта из обозревателя решений и перейдите во вкладку **Отладка**. Выберите действие при запуске, функцию **Не запускать, но произвести отладку моего кода при запуске**, и сохраните файл свойств. Вы можете использовать эту опцию, чтобы отладить приложение, которое вы не запускаете прямо из Visual Studio.

11. Для активации отладки без запуска приложения используйте кнопку **Начало отладки**.

12. Установите точку останова на **App.xaml.cs** после печати отладочных сообщений Debug.WriteLine.

13. Щёлкните по кнопке микрофона в своей панели задач для подготовки к запуску через голосовую команду.

14. Произнесите слова: " Hands-on Labs, launch". Cortana словами подтвердит, что открывает ваше приложение.

Примечание: Если ваши настройки региона, языка и распознавания речи будут установлены на другой поддерживаемый язык, для которого вы создали набор голосовых команд, вы можете использовать команды на другом языке. Если вы измените свои настройки региона, языка и речи, вам может понадобиться выйти из системы и снова зайти, чтобы изменение полностью вступило в силу.

15. Когда вы достигнете точки останова, откройте **окно вывода**, чтобы просмотреть сообщения об отладке. Вы увидите команду, которую распознала Cortana, а также фактически произнесенный текст. Они могут различаться в зависимости от того, насколько хорошо Cortana истолковала произнесенный вами текст. Используйте кнопку **Continue (Продолжить)** для возобновления запуска приложения из точки останова.

16. Ваше приложение запустится и перейдет к главной странице.

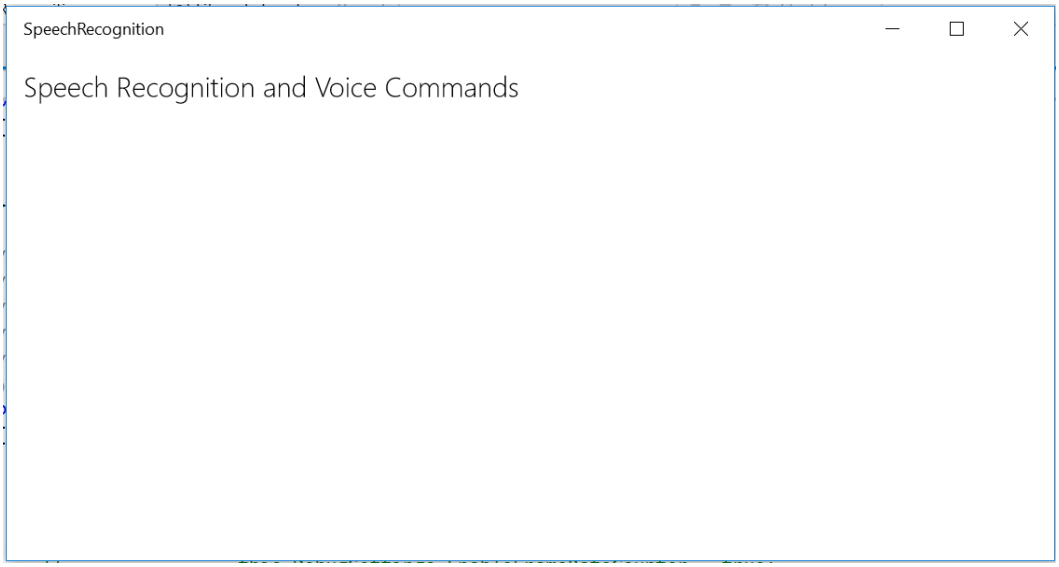

Рисунок 6
Приложение запускается через голосовую команду.

17. Закройте приложение и вернитесь в Visual Studio. Удалите точку останова из **App.xaml.cs**, и снимите галочку с опции **Do not launch, but debug my code when it starts (Не запускать, произвести отладку моего кода при активации)** в свойствах проекта. Сохраните файл.

Упражнение 2: Используйте голосовую команду, чтобы изменить вид приложения

В дополнение к запуску вашего приложения, голосовые команды могут взаимодействовать с содержимым приложения. В этом упражнении вы будете использовать голосовую команду для изменения фонового цвета приложения при его запуске.

Задача 1 – Установите цвет фона

В этой задаче вы зададите исходный цвет фона для приложения и атрибут **x:Name**, чтобы можно было обращаться к свойствам Grid.

1. Укажите **AliceBlue** как цвет фона для объкта **Grid** в **MainPage.xaml** и присвойте ему имя **Container**.

XAML

```
<Grid Background="AliceBlue" x:Name="Container" >
        <TextBlock Text="Speech Recognition and Voice Commands"
FontWeight="Light" FontSize="20" Margin="12" />
</Grid>
```

2. Создайте и запустите своё приложение на Локальном компьютере. Вы увидите заголовок страницы на светло-синем фоне.

Рисунок 7

Работа приложения с цветным фоном.

3. Завершите отладку и вернитесь в Visual Studio.

Задача 2 – Создайте голосовую команду, чтобы вызвать изменение цвета

В этой задаче вы создадите голосовую команду для изменения фонового цвета сетки на красный.

1. Откройте свой файл определения VoiceCommands.xml и добавьте новую команду. Присвойте команде имя **TurnRed.**

XML

```
<Command Name="TurnRed">
    <Example>turn red</Example>
    <ListenFor>turn red</ListenFor>
    <Feedback>My favorite color is red</Feedback>
    <Navigate />
</Command>
```

Примечание: Если вы поддерживаете дополнительные языки, добавьте эквивалентную команду в соответствующий набор команд.

2. Добавьте случай для команды **TurnRed** в оператор switch, обрабатывающий **voiceCommandName** в файле **App.xaml.cs**. В этот раз при переходе на главную страницу MainPage передайте параметр.

C#

```
switch (voiceCommandName)
{
    case "LaunchApp":
        frame.Navigate(typeof(MainPage));
        break;
    case "TurnRed":
        frame.Navigate(typeof(MainPage), "Red");
        break;
    default:
        break;
}
```

3. Откройте код MainPage и создайте переопределение **OnNavigatedTo** для обработки входящего параметра. Задайте фон Grid в соответствии с входящем параметром.

C#

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    if (e.Parameter.ToString() == "Red")
        Container.Background = new
SolidColorBrush(Windows.UI.Colors.DarkRed);

    base.OnNavigatedTo(e);
}
```

4. Щёлкнитесь правой кнопкой мыши на проекте в обозревателе решений и выберите **Развёртывание**.

5. Создайте и запустите своё приложение на Локальном компьютере для регистрации новой голосовой команды. Вы должны всё еще видеть синий фон. Закройте своё приложение.

6. Щёлкните по кнопке микрофона Cortana и проговорите команду: **" Hand-on Labs, turn red ".** Cortana ответит, что красный – её любимый цвет. При следующем запуске приложения вы увидите, что цвет фона заменен на красный.

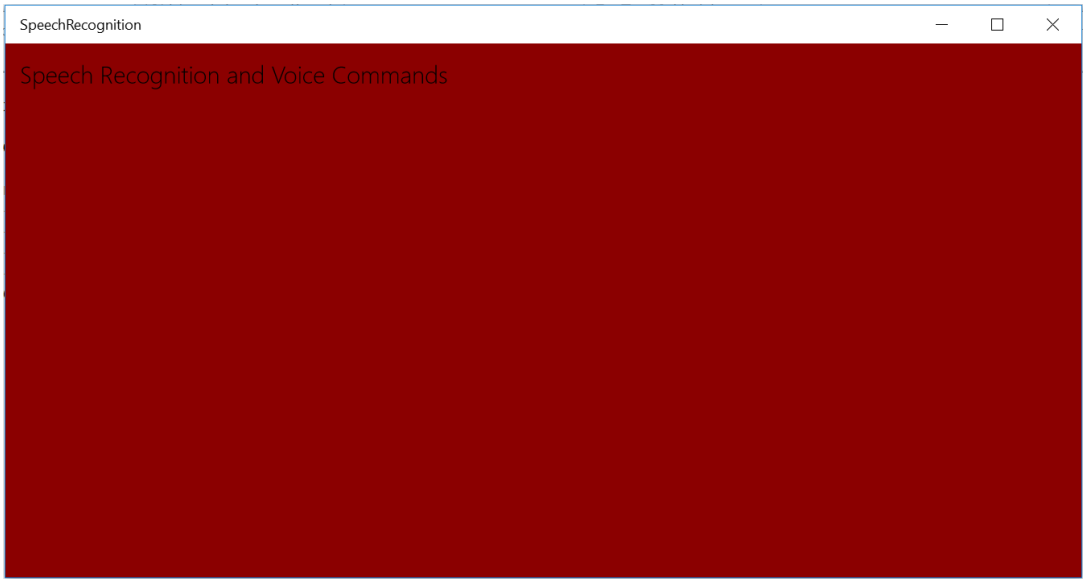

Рисунок 8
Приложение запускается через голосовую команду с красным фоном.

Примечание: Если вам нужно отладить код, который вы написали в этой задаче, используйте рассмотренный ранее метод с установкой опции **Do not launch, but debug my code when it starts (Не запускать, произвести отладку моего кода при активации)** из Упражнения 1: Задача 4.

7. Завершите отладку и вернитесь в Visual Studio.

Упражнение 3: Ответ на голосовую команду при помощи фоновой задачи

Голосовые команды могут активировать работу фоновых задач без запуска вашего приложения. Такая схема может быть полезной, когда вы хотите разрешить своим пользователям выполнять простые задачи, связанные с вашим приложением, непосредственно через Cortana без запуска приложения. В этом упражнении вы создадите компонент для ответа на вопрос пользователя через фоновую задачу.

Задача 1 – Создайте компонент времени выполнения Windows

Мы начнём с создания компонента времени выполнения Windows для фоновой задачи:

1. Щёлкните правой кнопкой мыши на названии решения в Обозревателе решений. Выберите **Добавить -> Новый Проект** и выберите тип проекта **компонент времени выполнения Windows** (Windows Универсальная).

2. Назовите проект **VoiceCommandService**.

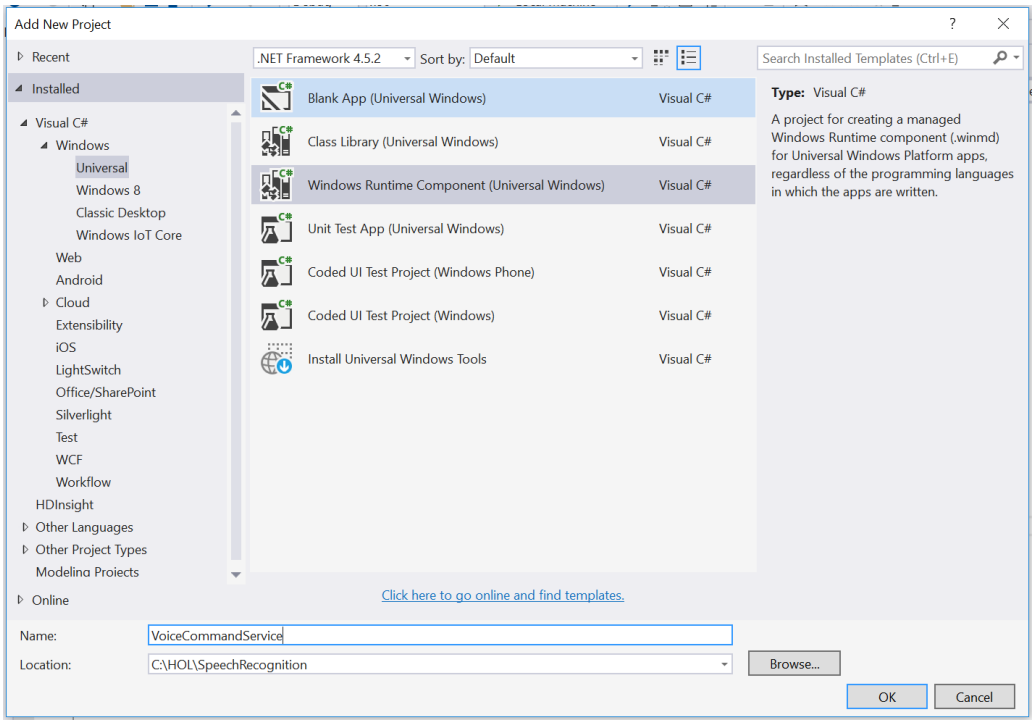

Рисунок 9

Добавьте Компонент Времени выполнения Windows.

3. Щёлкните правой кнопкой мыши по Class1.cs в Обозревателе решений и переименуйте его в **HolVoiceCommandService**. Если появится опция сменить имя проекта по всем ссылкам, выберите **"Да"**.

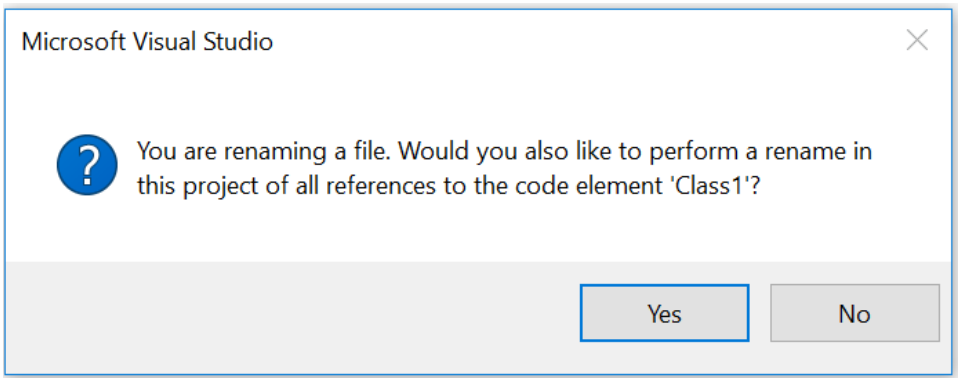

Рисунок 10

Переименуйте Class1.cs в HolVoiceCommandService.cs.

4. Вернитесь к проекту Распознавание речи. Щёлкните правой кнопкой мыши по разделу References (ссылки) и добавьте VoiceCommandService к списку ссылок.

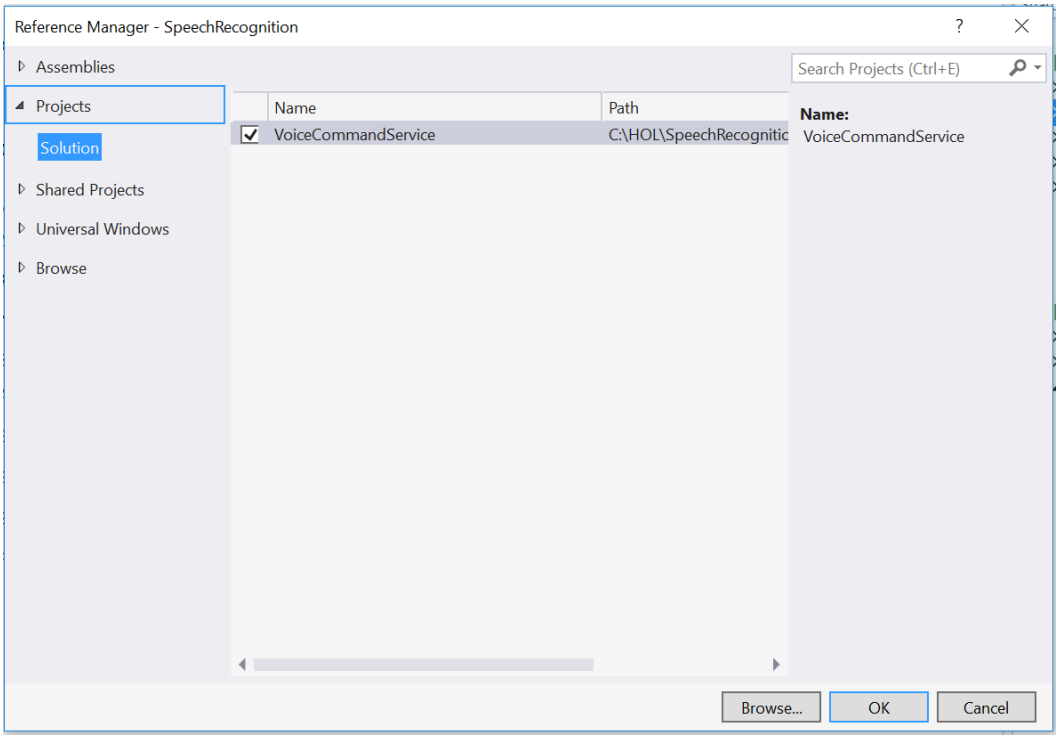

Рисунок 11

Ссылка на VoiceCommandService из проекта SpeechRecognition.

Задача 2 – Добавьте голосовую команду, ссылающуюся на VoiceCommandService

Голосовые команды, которые запускают фоновые задачи, отличаются от голосовых команд, которые вы создавали раньше. Вместо того, чтобы использовать элемент Navigate, команда задает пользовательский элемент VoiceCommandService , имеющий целевой атрибут с указанием на класс HolVoiceCommandService. В этой задаче вы создадите голосовую команду, а также предоставите ей дополнительные речевые опции. Вы можете добавить речевые опции в любые голосовые команды, независимо от того, используются ли они для запуска приложения на переднем плане или фоновой задачи.

1. В своём проекте SpeechRecognition добавьте к **VoiceCommands.xml** команду **SayHello**. Вы можете определять в команде многочисленные <**ListenFor>** элементы. В этом случае определите для Cortana два ключевых слова, оба из которых будут служить для запуска команды.

Примечание: Если вы поддерживаете дополнительные языки, добавьте эквивалентную команду в соответствующий набор команд.

XML

```
<Command Name="SayHello">
    <Example>say hello</Example>
    <ListenFor RequireAppName="BeforeOrAfterPhrase">How's it
going</ListenFor>
    <ListenFor RequireAppName="BeforeOrAfterPhrase">Say hello</ListenFor>
    <Feedback>Hold on, let me ask</Feedback>
    <VoiceCommandService Target="HolVoiceCommandService" />
</Command>
```

Примечание: Атрибут **RequireAppName="BeforeOrAfterPhrase"** придает гибкость и естественность словесному выражению ваших голосовых команд. Обе фразы "Лабораторный практикум, как дела?" И "Как дела, лабораторный практикум?" допустимы, если этот атрибут установлен в BeforeOrAfterPhrase. Для получения более подробной информации об опциях словесного выражения зайдите в документацию на ListenFor по ссылке https://msdn.microsoft.com/en-us/library/windows/apps/dn706593.aspx

Задача 3 – Зарегистрируйте сервис в манифесте приложения

Для фонового запуска HolVoiceCommandService, он должен быть зарегистрирован в манифесте приложения SpeechRecognition.

1. Откройте Package.appxmanifest в редакторе манифестов и перейдите на вкладку Declarations.

2. Используя выпадающее меню **Available Declarations**, выберите **App Service (Служба приложений)** и кликните **Добавить**, чтобы добавить службу в список поддерживаемых описаний.

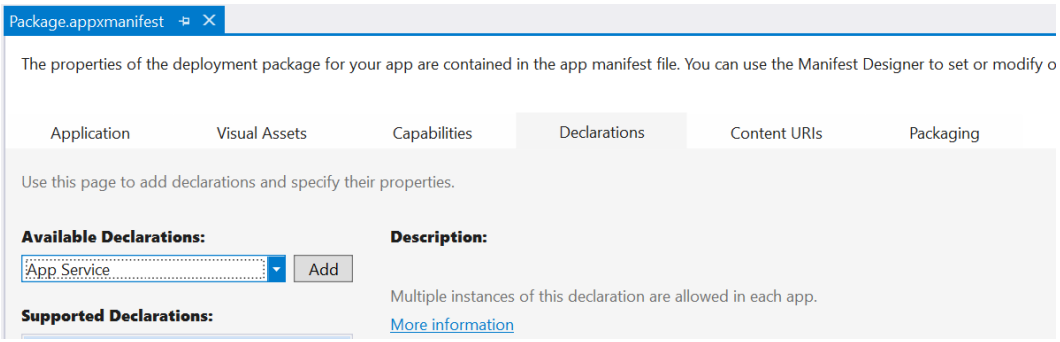

Рисунок 12

Добавьте Службу приложений в список поддерживаемых описаний.

3. В Свойствах манифеста сервисов приложений, установите **Name = HolVoiceCommandService**.

4. Установите свойство точки входа (Entry Point) в **VoiceCommandService.HolVoiceCommandService**.

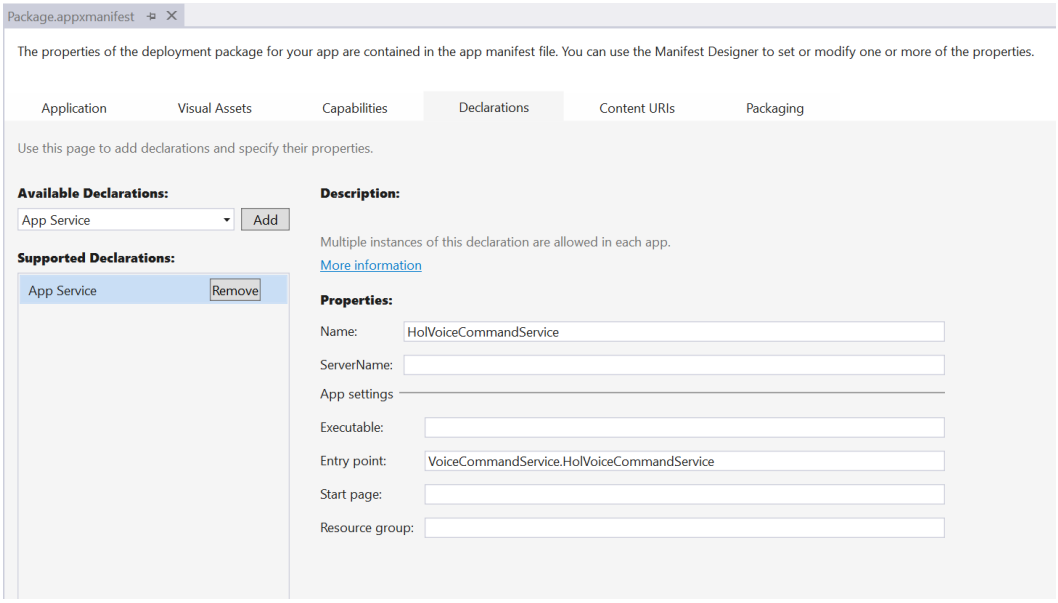

Рисунок 13

Зарегистрируйте Службу приложений в манифесте приложений.

Примечание: При регистрации службы приложений, свойство Name должно соответствовать имени класса в рамках вашего компонента, а не имени самого компонента. Мы дали классу HolVoiceCommandService имя, отличное от компонента WinRT, чтобы их было легче различать.

Это различие также важно при установке атрибута Target в определении голосовых команд, которое вы делали в задаче 2.

5. Закройте манифест, сохранив изменения.

Задача 4 – Обработайте входящую команду в сервисе

В этой задаче необходимо реализовать точку входа для всех скрытых (не приводящих к запуску приложения) голосовых команд, запущенных через Cortana. Фоновая задача, которую вы создадите, должна в течение 0.5 секунды предоставить ответ Cortana, и каждые 5 секунд должна сообщать о прогрессе – в противном случае она будет принудительно закрыта.

1. Откройте **HolVoiceCommandService.cs** и добавьте пространство имен Windows.ApplicationModel.Background.

C#

```
using Windows.ApplicationModel.Background;
```

2. Модифицируйте описание класса так, чтобы он реализовывал интерфейс **IBackgroundTask**. Добавьте определенный в этом интерфейсе метод **Run**

C#

```
public sealed class HolVoiceCommandService : IBackgroundTask
    {
        public void Run(IBackgroundTaskInstance taskInstance)
        {
            throw new NotImplementedException();
        }
    }
```

3. Объявите элемент класса **serviceDeferral** типа **BackgroundTaskDeferral**. В своём методе **Run** получите deferral для своего экземпляра задачи и сохраните его в serviceDeferral. Добавьте методы **OnVoiceCommandCompleted()** и **OnTaskCanceled()** для обработки завершения. На более позднем этапе вы оформите подписку на события **VoiceCommandCompleted**.

C#

```
public sealed class HolVoiceCommandService : IBackgroundTask
    {
        BackgroundTaskDeferral serviceDeferral;

        public void Run(IBackgroundTaskInstance taskInstance)
        {
            serviceDeferral = taskInstance.GetDeferral();

            taskInstance.Canceled += OnTaskCanceled;
        }
    }

    private void OnVoiceCommandCompleted(VoiceCommandServiceConnection
sender, VoiceCommandCompletedEventArgs args)
    {
        if (this.serviceDeferral != null)
        {
            this.serviceDeferral.Complete();
        }
```

```
    }

    private void OnTaskCanceled(IBackgroundTaskInstance sender,
BackgroundTaskCancellationReason reason)
    {
        System.Diagnostics.Debug.WriteLine("Задача отменена, очистить");
        if (this.serviceDeferral != null)
        {
            //Завершить отсрочку службы
            this.serviceDeferral.Complete();
        }
    }
```

4. Добавить пространства имен **Windows.ApplicationModel.AppService** и **Windows.ApplicationModel.VoiceCommands.**

C#

```
using Windows.ApplicationModel.AppService
using Windows.ApplicationModel.VoiceCommands;
```

5. Добавьте ссылку на подключение к службе голосовых команд.

C#

```
public sealed class HolVoiceCommandService : IBackgroundTask
{
    VoiceCommandServiceConnection voiceServiceConnection;

    BackgroundTaskDeferral serviceDeferral;
```

Примечание: Подключение к службе сохраняется в течение всего периода работы с Cortana.

6. Проверьте, соответствует ли имя указанному при регистрации службы приложений из манифеста приложений. Если да, то установите соединение со службой Cortana, используя блок try-catch для обнаружения проблем.

C#

```
taskInstance.Canceled += OnTaskCanceled;

var triggerDetails = taskInstance.TriggerDetails as
AppServiceTriggerDetails;

if (triggerDetails != null && triggerDetails.Name ==
"HolVoiceCommandService")
{
    try
    {
        voiceServiceConnection =

VoiceCommandServiceConnection.FromAppServiceTriggerDetails(
                            triggerDetails);
```

```
    voiceServiceConnection.VoiceCommandCompleted +=
OnVoiceCommandCompleted;
    }
    catch (Exception ex) {
    System.Diagnostics.Debug.WriteLine ("Обработка голосовой команды
потерпела неудачу" + ex.ToString());
    }
}
```

Примечание: Подписка на событие VoiceCommandCompleted находится именно в этом фрагменте кода, поскольку оно должно произойти после установки voiceServiceConnection.

Если вам нужно отладить этот код, используйте метод, описанный выше, и установите в вашем методе run точку останова.

7. Используйте await для ожидания входящей голосовой команды и создайте переключатель для обработки команды **SayHello**. Добавьте ключевое слово **async** к заголовку метода Run.

C#

```
try
{
    voiceServiceConnection =

VoiceCommandServiceConnection.FromAppServiceTriggerDetails(
                            triggerDetails);

    voiceServiceConnection.VoiceCommandCompleted +=
OnVoiceCommandCompleted;

    VoiceCommand voiceCommand = await
voiceServiceConnection.GetVoiceCommandAsync();

    switch (voiceCommand.CommandName)
    {
        case "SayHello":
            break;
        default:
            break;
    }
}
```

8. Если входящая голосовая команда будет **SayHello**, создайте ответное сообщение и задайте дисплейное и речевое сообщения, которые Cortana передаст обратно пользователю.

C#

```
switch (voiceCommand.CommandName)
{
    case "SayHello":
        var userMessage = new VoiceCommandUserMessage();
```

```
        userMessage.DisplayMessage = "Hello!";
        userMessage.SpokenMessage = "Your app says hi. It is having a great
time.";
        var response = VoiceCommandResponse.CreateResponse(userMessage);
        await voiceServiceConnection.ReportSuccessAsync(response);
        break;
    default:
        break;
}
```

9. Сохраните HolVoiceCommandService.

Задача 5 – Начните работу со своим приложением через фоновую задачу

Теперь, когда вы настроили голосовую команду, фоновую задачу и зарегистрировали службу приложений, самое время попробовать, как это работает.

1. Запустите проект SpeechRecognition на Локальном компьютере, чтобы зарегистрировать самую последнюю версию файла определения голосовых команд.

2. Закройте приложение.

3. Используйте интерфейс Cortana для того, чтобы проговорить один из вариантов голосовой команды SayHello:

   - “Hands-on labs, how’s it going?”
   - “How’s it going, Hands-on labs?”
   - Hands-on labs, say hello.”
   - “Say hello, Hands-on labs.

4. Cortana покажет изображение вашего приложения и дисплейное сообщение. Одновременно она проговорит речевое сообщение, которое вы выбрали в фоновой задаче.

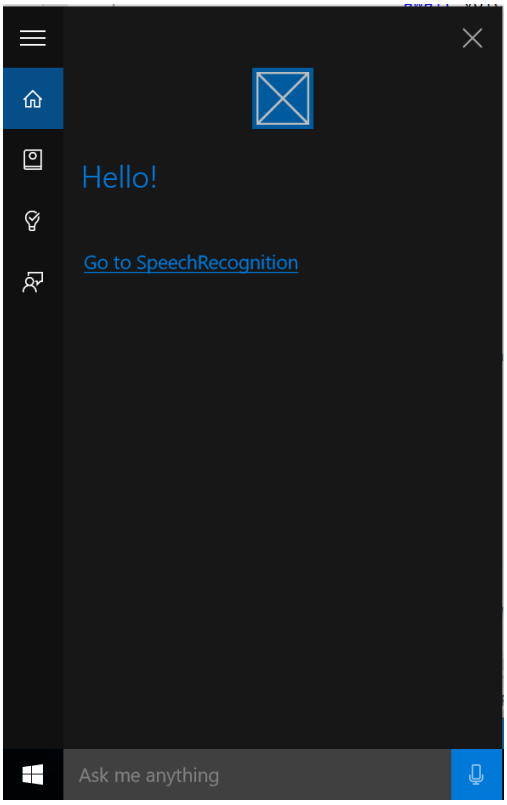

Рисунок 14

Cortana возвращает сообщение из фоновой задачи приложения.

Примечание: Чтобы отладить свою фоновую задачу, используйте метод аналогичный тому, который вы использовали для отладки своих голосовых команд. Щёлкните правой кнопкой мыши на проекте SpeechRecognition в Обозревателе решений и откройте редактор **Свойств**. На вкладке **Отладка** отметьте опцию **Не запускать, произвести отладку моего кода при активации**. Сохраните файл свойств и используйте кнопку пуска отладки для активации отладки. Установите точку останова в HolVoiceCommandService и запустите приложение, используя команду SayHello.

5. Остановите отладку и вернитесь в Visual Studio.

Краткий обзор

Голосовые команды – это важная часть взаимодействия Windows 10 с пользователем. В этой работе вы создали файл определения голосовых команд и исследовали структуру VCD. Вы добавили команды запуска приложения, взаимодействия с ним и получения ответа из фоновой задачи. Вы также научились регистрировать и различать голосовые команды.