

Создание аудио проигрывателя с помощью Microsoft Visual Studio и библиотеки Naudio

Маринчук Александр Сергеевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В данной статье был создан аудио проигрыватель с помощью Microsoft Visual Studio и библиотеки Naudio. Добавлены функции проигрывания, паузы, стоп и изменения громкости для выбранного аудио файла в формате mp3, который наиболее распространён среди форматов хранения данных со сжатием.

Ключевые слова: Naudio, Microsoft Visual Studio, музыка, аудио формат.

Create an audio player using Microsoft Visual Studio and the Naudio library

Marinchuk Alexander Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article, an audio player was created using Microsoft Visual Studio and the Naudio library. Added functions for playing, pausing, stopping and changing the volume for the selected audio file in mp3 format, which is most common among data storage formats with compression

Keywords: Naudio, Microsoft Visual Studio, music, audio format.

1. Введение

1.1 Актуальность исследования

Звуки сопровождают человека в повседневной жизни с самого детства. Общение, прослушивание музыки, аудио сообщения в различных мессенджерах – всё это проявления звуков в различных его видах. Трудно представить обыденный день без восприятия какой-либо информации через звуковые сигналы. Но мало кто задумывается о том, как музыка или аудио сообщения доносится до нас через информационные технологии. Проигрыватель стал настолько привычной вещью, что его отсутствие в какой либо социальной сети или как самостоятельного приложения воспринимается отрицательно на подсознательном уровне. Именно поэтому очень важно иметь под рукой данный инструмент и использовать его в нужных целях.

1.2 Обзор исследований

В статье В. Н. Зимина и П. В. Федосова описывается разработка веб-приложений на основе технологии AJAX в среде Microsoft Visual Studio [1]. Статья Д. Д. Галиуллина посвящена поэтапной разработке мультимедийного плеера на языке C#, рассматриваются вопросы проектирования и программной реализации плеера [2]. Рассмотрели основные особенности визуального программирования в программе в среде Microsoft Visual Studio, а также представили особенности разработки дополнительного курса для инженеров электронной техники N. Wilson и D. Murzin [3]. В. А. Михайлов, А. Ш. Исянова и Т. А. Михайлова в своей статье описали подход к управлению макросами файла Microsoft Excel с использованием приложения, разработанного в среде Visual Studio [4]. Д. В. Галенин посвятил свою статью нахождению корней заданного уравнения при помощи Microsoft Visual Studio с дальнейшим выбором наиболее оптимального метода решения [5].

1.3 Цель исследования

Целью данной статьи является создание аудио проигрывателя с помощью Microsoft Visual Studio и библиотеки NAudio.

2. Методы исследования

Для разработки аудио проигрывателя был выбран инструмент Microsoft Visual Studio, который является простым и удобным решением для создания приложений.

Microsoft Visual Studio — это узкоспециализированная линейка программных продуктов от компании Microsoft, которая включает в себя интегрированную среду разработки программного обеспечения и множество других средств для разработчиков [6].

Для начала работы следует скачать и установить Microsoft Visual Studio с официального сайта (рис. 1).

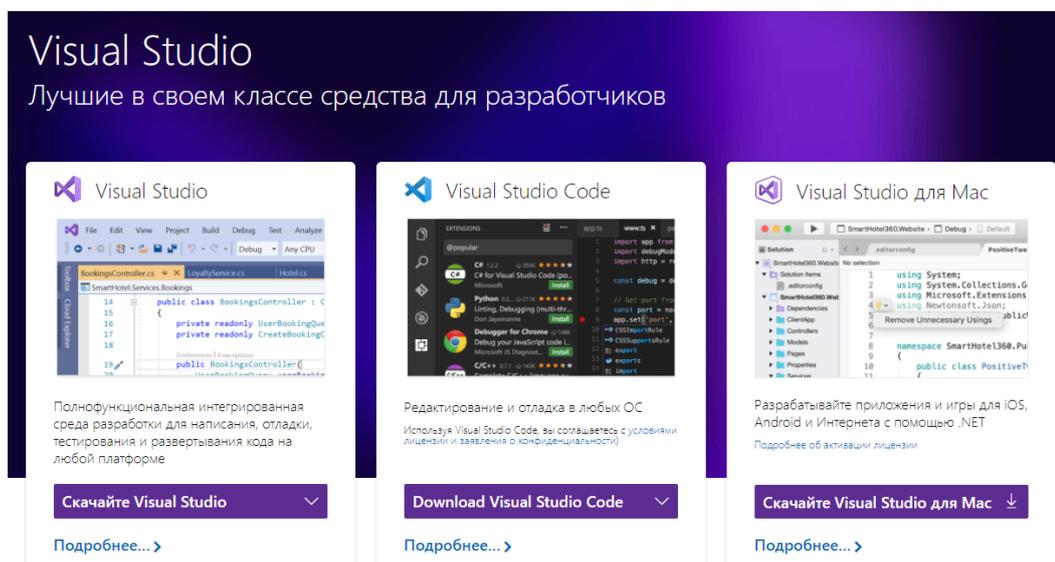


Рисунок 1 – Скачивание приложения

После установки программы следует создать новый проект (рис. 2).



Рисунок 2 – Создание проекта

Далее добавим на созданную форму 3 кнопки, которые будут отвечать за проигрывание аудио файла, его паузу и полную остановку, подразумевающую переход к началу проигрывания. Для этого перейдем во вкладку панель элементов, найдем через поиск button и перетащим данный элемент на форму (рис. 3).

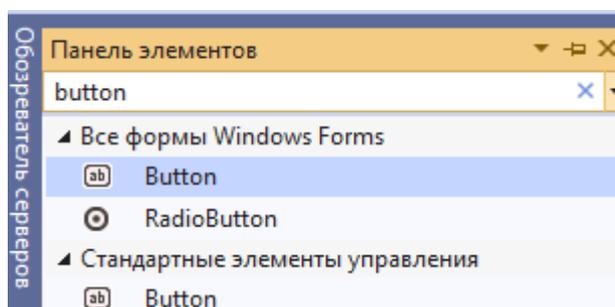


Рисунок 3 – Добавление компонента button

Сотрем текст с кнопок в свойстве Text и загрузим в свойство BackgroundImage изображения, характеризующие функции данных кнопок (рис. 4).

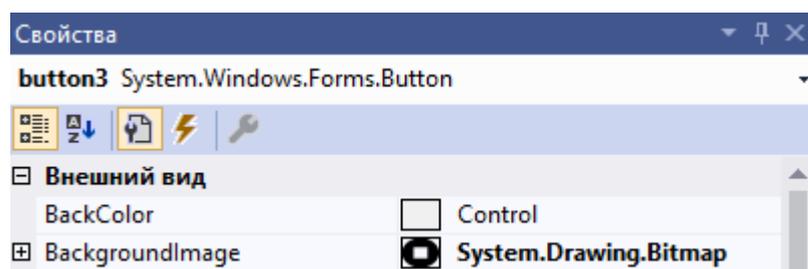


Рисунок 4 – Свойство BackgroundImage

Теперь добавим элемент TrackBar, у которого изменим значения свойств Minimum и Maximum на 0 и 100 соответственно. Также добавим элементы Label и TextBox (рис. 5).

Поведение	
AllowDrop	False
AutoSize	True
ContextMenuStrip	(нет)
Enabled	True
LargeChange	1
Maximum	100
Minimum	0

Рисунок 5 – Свойства TrackBar

И последним добавляемым элементом станет MenuStrip, который понадобится для загрузки аудио файлов (рис. 6).

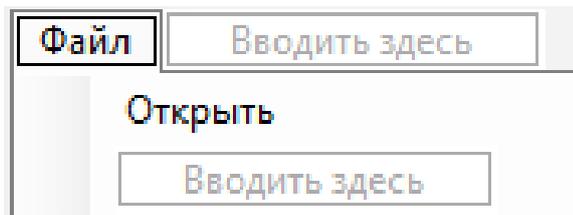


Рисунок 6 – Элемент MenuStrip

Теперь подключим необходимую библиотеку NAudio, которая служит для работы с аудио файлами. Перейдем на вкладку обозреватель решений, кликнем правой кнопкой мыши по нашему проекту и перейдем на управление проектами NuGet. Там необходимо ввести в поиске библиотеку NAudio и нажать кнопку установить (рис. 7).

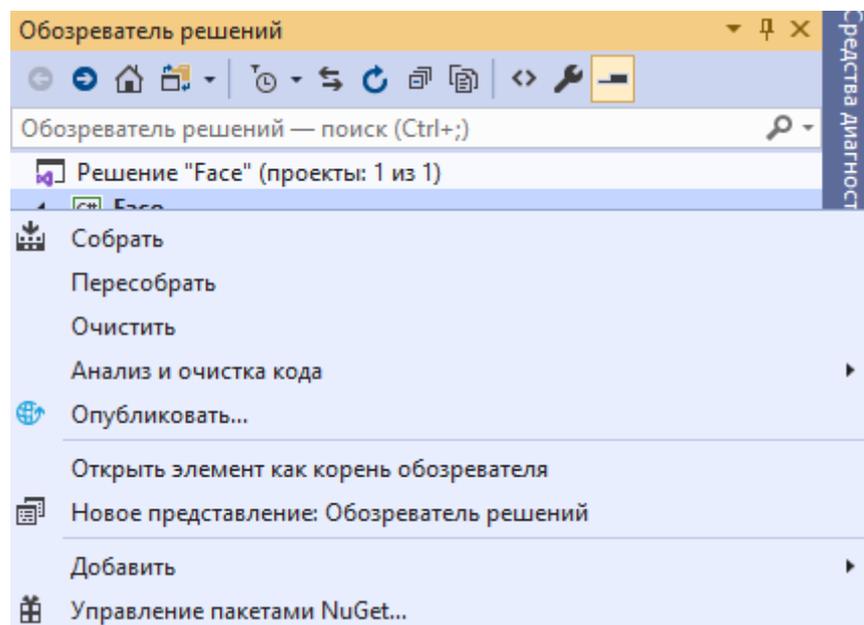


Рисунок 7– Скачивание библиотеки Emgu CV

Далее перейдем на страницу редактирования кода, дважды щелкнув на главную форму. Здесь нам необходимо подключить скачанную библиотеку двумя строчками (рис. 8).

```
using NAudio.Wave;  
using NAudio.Wave.SampleProviders;
```

Рисунок 8 – Подключение библиотеки

Затем пропишем методы `WaveOutEvent` и `AudioFileReader`, которые служат для выбора устройства вывода звука и его воспроизведения соответственно. Также добавим переменную `filename` для хранения полного названия файла (рис. 9).

```
private WaveOutEvent outputDevice;  
private AudioFileReader audioFile;  
string filename = string.Empty;
```

Рисунок 9 – Методы и переменная

Далее создадим класс `OpenFileMedia`, который будет отвечать за открытие аудио файлов, и укажем необходимые свойства (рис. 10).

```
private void OpenFileMedia()  
{  
    OpenFileDialog openfile = new OpenFileDialog()  
    {  
        Filter = "Mp3|*.mp3",  
        Multiselect = false,  
        ValidateNames = true  
    };  
  
    if (openfile.ShowDialog() == DialogResult.OK)  
    {  
        if (outputDevice != null || audioFile != null)  
        {  
            outputDevice.Dispose();  
            audioFile.Dispose();  
            outputDevice = new WaveOutEvent();  
            audioFile = new AudioFileReader(filename);  
            outputDevice.Init(audioFile);  
        }  
        filename = openfile.FileName;  
        string a = System.IO.Path.GetFileNameWithoutExtension(@"C:\aq\bw\ce\" + openfile.FileName);  
        textBox1.Text = a;  
        if (outputDevice == null)  
        {  
            outputDevice = new WaveOutEvent();  
            audioFile = new AudioFileReader(filename);  
            outputDevice.Init(audioFile);  
        }  
    }  
}
```

Рисунок 10 – Класс `OpenFileMedia`

В случае успешной загрузки файла будем с помощью условия проверять текущие значения `outputDevice` и `audioFile`. Если они не пустые, то будем очищать их и пересоздавать заново, а иначе просто создавать. Также

занесем название загруженного файла в переменную filename и присвоим отфильтрованное название элементу TextBox.

Затем напишем обработчик нажатия кнопки play. Для этого дважды щелкнем на кнопку ЛКМ и внутри пропишем следующий код (рис. 11).

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        outputDevice.Play();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        OpenFileMedia();
    }
}
```

Рисунок 11 – Обработчик нажатия кнопки play

В случае нажатия кнопки play будем проигрывать выбранный аудио файл, а при возникновении ошибки будем выводить сообщение в отдельном окне с ее описанием.

Теперь напишем обработчики нажатия кнопок pause и stop похожим образом (рис. 12).

```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        outputDevice.Pause();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        OpenFileMedia();
    }
}

ссылка:1
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        outputDevice.Stop();
        audioFile.Position = 0;
    }
    catch (Exception ex)
    {
        MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        OpenFileMedia();
    }
}
```

Рисунок 12 – Код обработки нажатия кнопок pause и stop

Следующим шагом будет написание кода для ползунка изменения громкости. Для этого будем передавать текущее значение ползунка методу Volume, который в соответствии с полученным значением будет изменять громкость (рис. 13).

```
private void trackBar1_Scroll(object sender, EventArgs e)
{
    try
    {
        trackBar1.Scroll += (s, a) => audioFile.Volume = trackBar1.Value/100f;
    }
    catch (Exception ex)
    {
        MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        OpenFileMedia();
    }
}
```

Рисунок 13 – Обработчик TrackBar

И последним шагом станет вызов класса OpenFileMedia при нажатии по кнопке «Открыть» элемента MenuStrip (рис. 14).

```
ссылка: 1
private void открытьToolStripMenuItem1_Click(object sender, EventArgs e)
{
    OpenFileMedia();
}
```

Рисунок 14 – Обработчик MenuStrip

В итоге получим главную форму следующего вида (рис. 15).

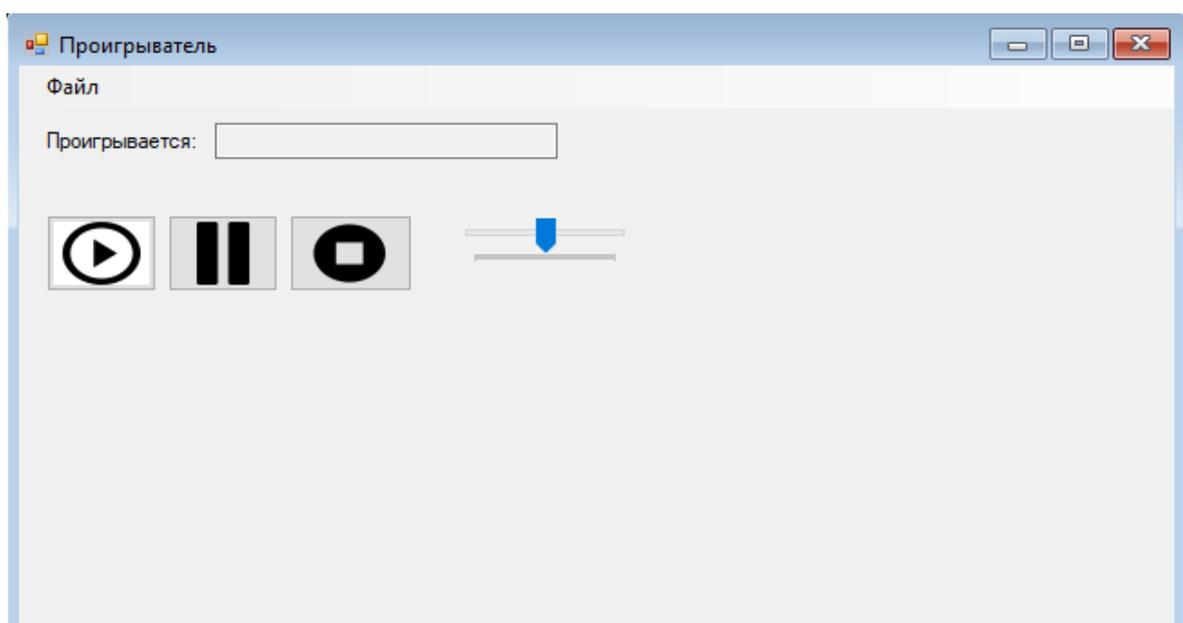


Рисунок 15 – Главная форма

Попробуем запустить аудио проигрыватель и выбрать любой mp3 файл для его прослушивания. Название файла должно отобразиться в TextVox (рис. 16).

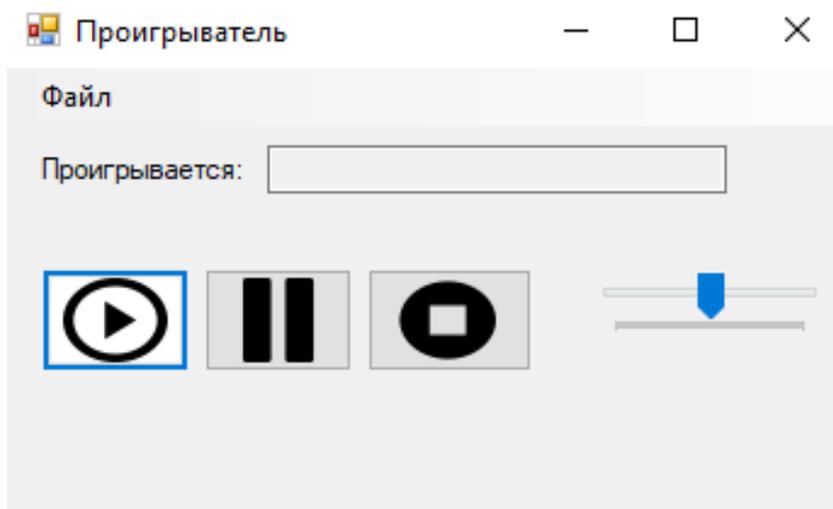


Рисунок 16 – Готовое приложение

Выводы

Вообразить современную жизнь без прослушивания песен или иных аудио файлов, несомненно, сложно. Люди обмениваются или делятся десятками тысяч аудио файлов в день и создание удобных и качественных инструментов для их прослушивания является одной из важных задач по обеспечению людей качественным звуком.

В данной статье были рассмотрены инструмент Microsoft Visual Studio и библиотека Naudio для создания аудио проигрывателя. В будущем планируется добавление функции перемотки аудио файла, а также расширение списка поддерживаемых форматов.

Библиографический список

1. Зимин В. Н., Федосов П. В. Разработка веб-приложений на основе технологии AJAX в среде Microsoft Visual Studio // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2008. №51. С. 103-107.
2. Галиуллин Д. Д. Применение языка программирования C# для разработки программных продуктов // Аллея науки. 2019. №1. С. 1016-1019.
3. Wilson N., Murzin D. Visual design of programs in Microsoft Visual Studio Environment // Достижения и перспективы инноваций и технологий. Керчь: ФГБОУ ВО «Керченский государственный морской технологический университет», 2019. С. 160-166.
4. Михайлов В. А., Исянова А. Ш., Михайлова Т. А. Использование возможностей среды Visual Studio для разработки приложения управления макросами файла Microsoft Excel // Актуальные проблемы науки и

- образования в современном вузе. Уфа: Башкирский государственный университет, 2019. С. 102-106.
5. Галенин Д. В. Применение Microsoft Visual Studio для решения уравнений // Теоретические и практические аспекты развития современной науки. Нефтекамск: Научно-издательский центр "Мир науки" (ИП Вострецов Александр Ильич), 2017. С. 18-23.
 6. Microsoft Visual Studio // Википедия URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата обращения: 27.01.2020).
 7. Naudio // Github URL: <https://github.com/naudio/NAudio> (дата обращения: 27.01.2020).