ЛАБОРАТОРНАЯ РАБОТА №2 (ЗАДАНИЕ 1)

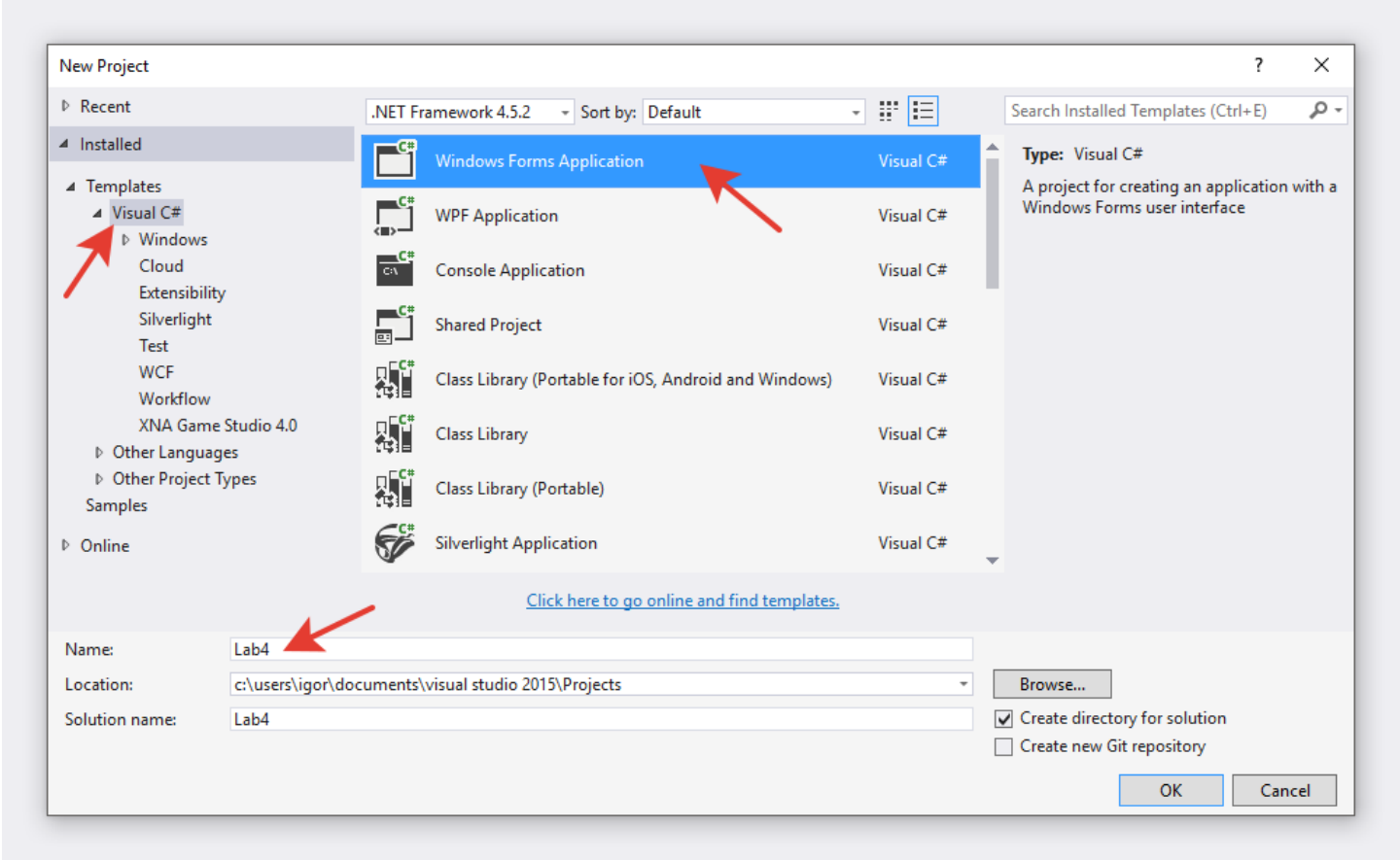
Проигрывание файлов формата WMA. Получение метаданных.

ВВЕДЕНИЕ

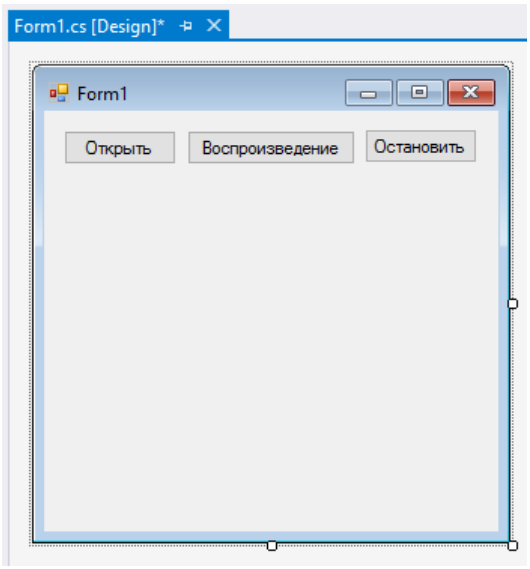
Эта лабораторная работа посвящена изучению базовых навыков работы с мультимедиа при помощи языка C#. Для выполнения лабораторной работы учащимся потребуется один файл формата Windows Media Audio. Файл можно подготовить самостоятельно или взять из сопровождения к лабораторной работе.

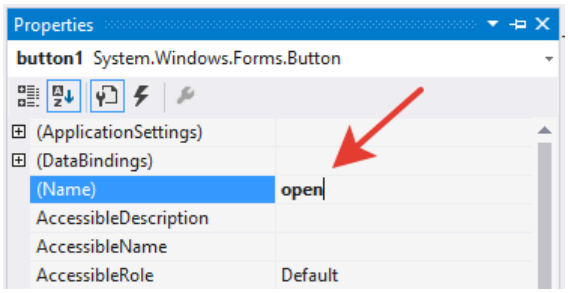
СОЗДАНИЕ НОВОГО ПРОЕКТА C#

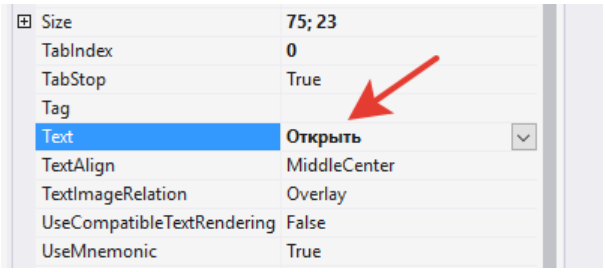
Создадим новый проект Windows Forms. По аналогии с лабораторной работой №2 выберите пункты меню **Файл – Новый – Проект** (**File – New – Project**). В открывшемся диалоговом окне выберите пункт **приложение Windows Forms** (**Windows Forms Application**)



Откроется редактор форм. Добавьте на форму три кнопки. В свойствах первой кнопки установите имя как «**open**», а текст как «**Открыть**». Для второй кнопки задайте имя «**play**» и текст «**Воспроизведение**». Для третьей соответственно укажите имя «**stop**» и текст «**Остановить**».







Дважды щелкните по кнопке **Открыть**. Среда Visual Studio сгенерирует обработчик события нажатия на кнопку. В нашем случае это метод **open_Click**, расположенный внутри класса **Form1**.

ОТКРЫТИЕ ФАЙЛА

Пропишем внутри этого метода код, открывающий стандартный диалог выбора файла.

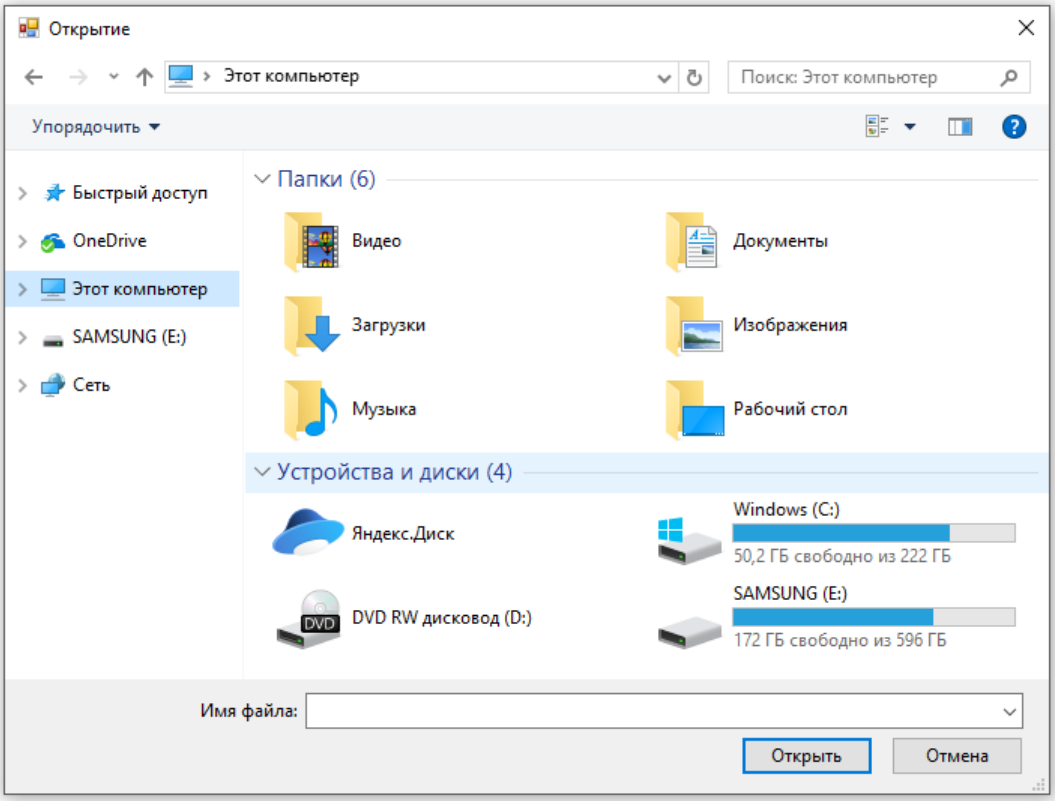
В библиотеке System.Windows.Forms уже имеется класс, позволяющий осуществить поиск и выбор файла, он называется OpenFileDialog. Создадим объект этого класса внутри метода **open_Click** :

```
private void open_Click(object sender, EventArgs e) {
    var dialog = new OpenFileDialog();
```

Показать диалоговое окно можно при помощи вызова метода ShowDialog:

```
dialog.ShowDialog();
```

После выполнения этой команды откроется диалоговое окно:



В зависимости от предпринятых действий метод **ShowDialog** вернет одно из нескольких значений:

- **DialogResult.OK** – была нажата кнопка **Открыть**.
- **DialogResult. Cancel** – была нажата кнопка **Отмена** или диалоговое окно было закрыто.
- **DialogResult.Abort, DialogResult.Retry, DialogResult. Ignore, DialogResult. Yes, DialogResult.No** – эти возвращаемые значения используются в других видах диалогов и к **OpenFileDialog** неприменимы.

Если мы хотим получить правильное имя файла, нам необходимо проверить, что была нажата кнопка **Открыть**. Делается это следующим образом:

```
if (dialog.ShowDialog() == DialogResult.OK) {
}
```

Настроим диалог на поиск файлов формата WMA. Укажем начальный каталог:

```
dialog.InitialDirectory = "c:\\";
```

Настроим фильтр файлов:

```
dialog.Filter = "Windows Media Audio (*.wma)|*.wma|All files (*.*)|*.*";
dialog.FilterIndex = 1;
```

Теперь, при нажатии на кнопку открыть у нас появится диалоговое окно, начальным каталогом будет корень диска **C**, а фильтр файлов будет установлен как ***.wma**.

Имя найденного файла можно получить при помощи свойства **FileName** объекта **dialog**:

```
var name = dialog.FileName;
```

К сожалению, объект **dialog** является локальным для метода **open_Click**, поэтому для передачи имени в другие обработчики нам потребуется добавить в класс **Form1** новое поле. Пропишите поле **filename** типа **string** сразу после заголовка класса Form1:

```
public partial class Form1 : Form {
    string filename;
```

Присвоим этому полю значение свойства **FileName**:

```
if (dialog.ShowDialog() == DialogResult.OK) {
    filename = dialog.FileName;
}
```

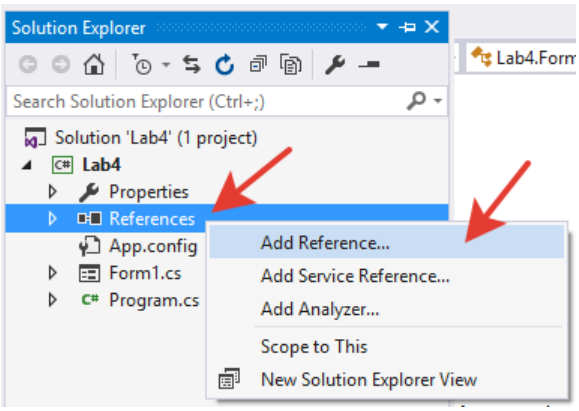
Теперь мы может использовать это имя в других методах класса **Form1**.

ВОСПРОИЗВЕДЕНИЕ

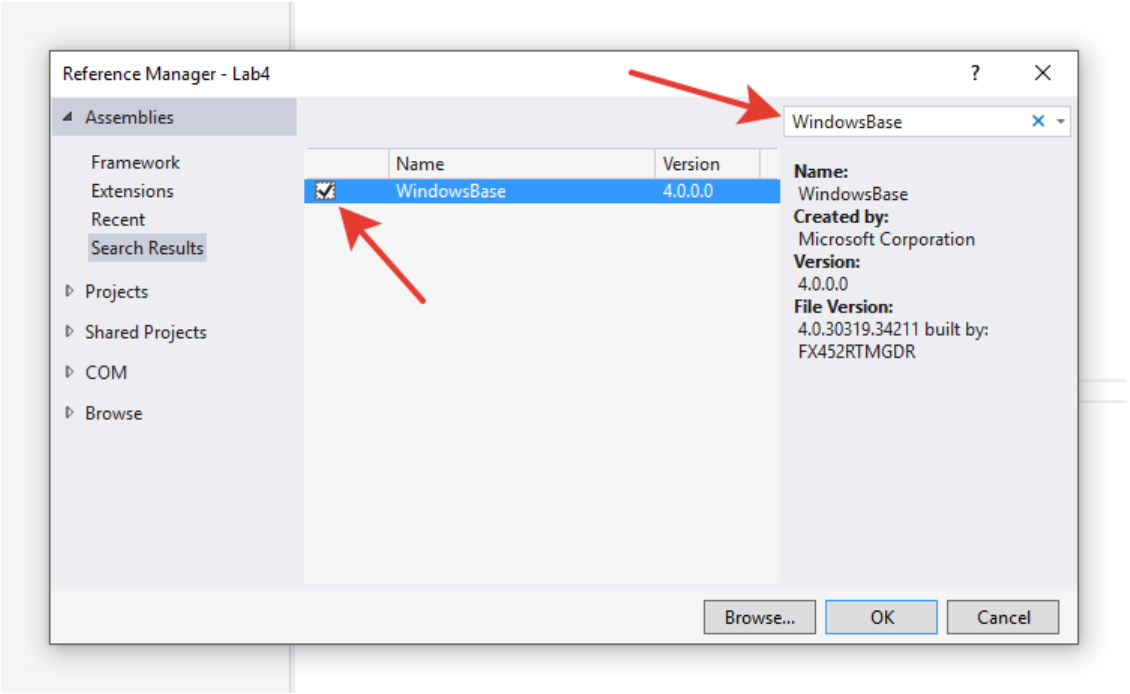
В редакторе форм дважды щелкните на кнопку **Воспроизведение**. Среда сгенерирует нам метод **play_Click**. Реализуем в этом методе проигрывание файла **filename**.

Нам потребуется подключить к проекту несколько стандартных библиотек.

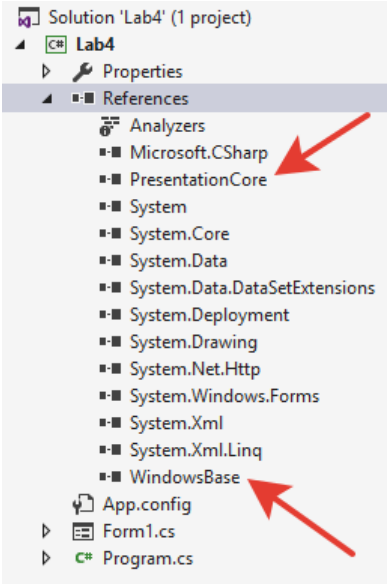
Откройте **Обозреватель решений** (**Solution Explorer**) и найдите узел **Ссылки** (**References**). Щелкните на этом узле правой кнопкой и выберите пункт выпадающего меню **Добавить ссылку** (**Add Reference**).



В открывшемся диалоговом окне воспользуйтесь поиском и поставьте галочки напротив сборок **PresentationCore** и **WindowsBase**.



Добавленные сборки должны появится в узле **Ссылки** (**References**):



Теперь у нас появился доступ к пространству имен **Windows.System.Media**. Добавьте в начало файла новую директиву **using**:

```
using System.Windows.Media;
```

Для работы со звуком мы будем использовать класс **MediaPlayer**. Пропишем для него объект внутри класса **Form1**:

```
public partial class Form1 : Form {
    MediaPlayer player = new MediaPlayer();
```

Нам не требуется передавать какие-либо параметры объекту класса **MediaPlayer**, поэтому мы создаем его прямо при инициализации класса **Form1**.

Перед проигрыванием звука нам необходимо его загрузить. За это отвечает метод **Open** объекта **player**. Этот метод принимает один параметр – адрес файла в формате **URI**.

Для создания адреса воспользуемся классом **Uri**:

```
var uri = new Uri("file://" + filename);
```

Мы генерируем адрес на основе значения, полученного из диалога **OpenFileDialog**, поэтому перед проигрыванием музыкального файла вам будет необходимо его найти и открыть.

Загружаем файл:

```
player.Open(uri);
```

И запускаем его проигрывание при помощи метода Play:

```
player.Play();
```

Если все было сделано правильно, то при нажатии на кнопку **Воспроизведение** вы должны услышать звук.

ПРЕРЫВАНИЕ ВОСПРОИЗВЕДЕНИЯ

В редакторе форм дважды щелкните на кнопке **Остановить**. Среда **Visual Studio** сгенерирует нам метод **stop_Click**.

Для прерывания воспроизводимой композиции достаточно вызывать метод **Stop** объекта **player**:

```
player.Stop();
```

После того мы можем закрыть загруженный файл:

```
player.Close();
```

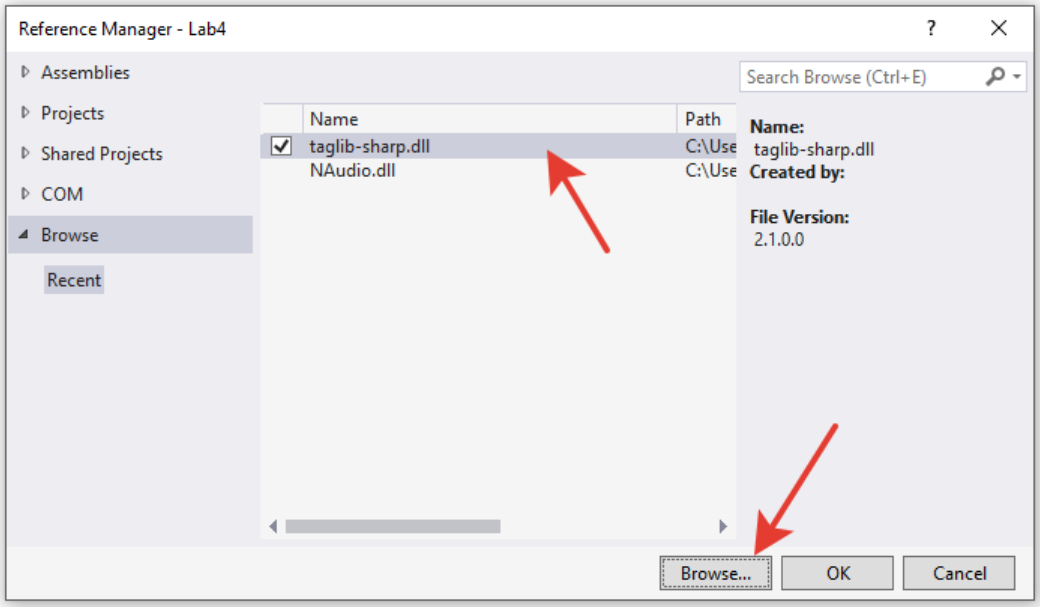
Добавьте этот код в тело метода **stop_Click**.

ЧТЕНИЕ ТЕГОВ

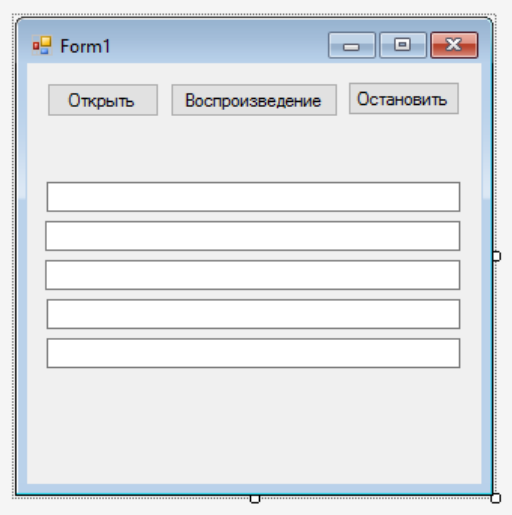
Давайте реализуем чтение тегов аудио файла. Возможности стандартных библиотек языка C# в этом плане довольно ограничены, поэтому мы воспользуемся библиотекой **TagLib**. Нам понадобится версия для языка C#, скачать которую можно по адресу http://download.banshee.fm/taglib-sharp/2.1.0.0/taglib-sharp-2.1.0.0-windows.zip

Найдите файл **taglib-sharp.dll** и скопируйте его в каталог с файлами лабораторной работы №4.

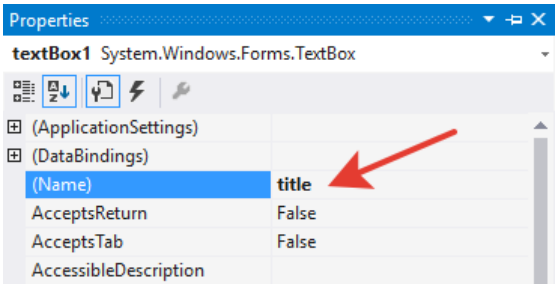
Откройте **Обозреватель решений** (**Solution Explorer**) и найдите узел **Ссылки** (**References**). Щелкните на этом узле правой кнопкой и выберите пункт выпадающего меню **Добавить ссылку** (**Add Reference**). В открывшемся диалоговом окне нажмите кнопку **Открыть** (**Browse**) и найдите скопированный файл dll-библиотеки **TagLib**.



Подключение завершено. Теперь вернитесь в **редактор форм** и добавьте в наше приложение **пять** элементов управления **TextBox**:



В панели свойств задайте им следующие имена (name): **title**, **artist**, **album**, **year** и **comment**.



Теперь мы можем прочитать информацию из тегов. Для этого нам потребуется объект класса **File** из пространства имен **TagLib**. Объект создается при помощи вызова статического метода **Create**:

```
var file = TagLib.File.Create(filename);
```

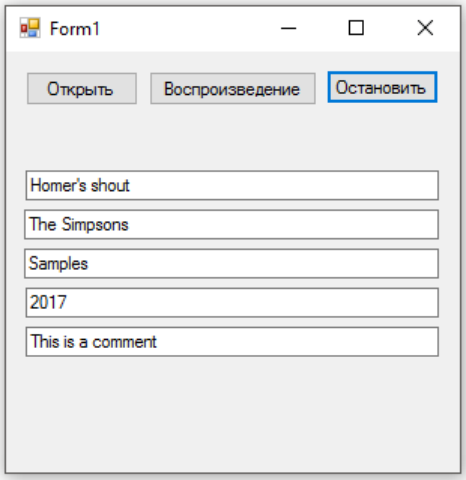
Теперь мы можем используя свойство **Tag** объекта **file** получить информацию о названии трека, исполнителе, альбоме и т.п.

```
title.Text = file.Tag.Title;
artist.Text = file.Tag.FirstPerformer;
album.Text = file.Tag.Album;
year.Text = file.Tag.Year.ToString();
comment.Text = file.Tag.Comment;
```

Этот код необходимо вставить в метод open_Click сразу после получения имени медиа файла:

```
if (dialog.ShowDialog() == DialogResult.OK) {
    filename = dialog.FileName;

    var file = TagLib.File.Create(filename);
    ...
```

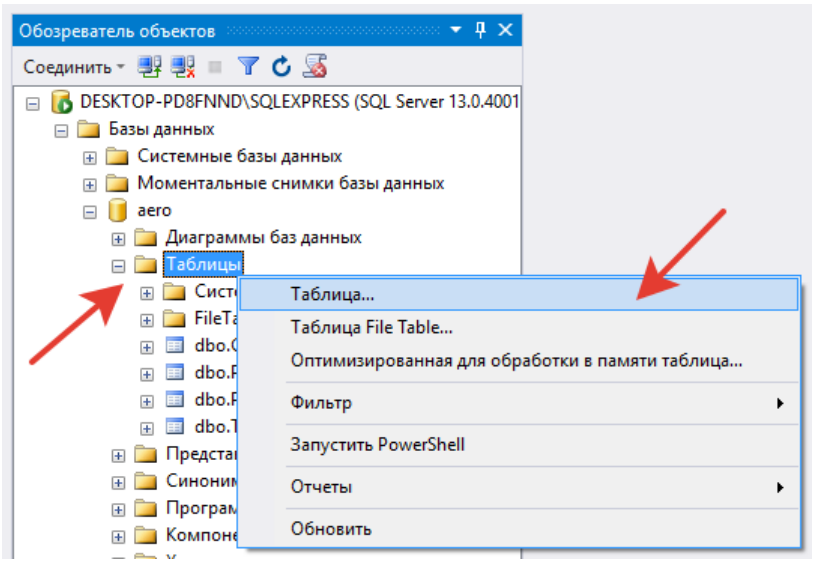
После запуска программы и открытия аудио файла мы получаем следующий результат:



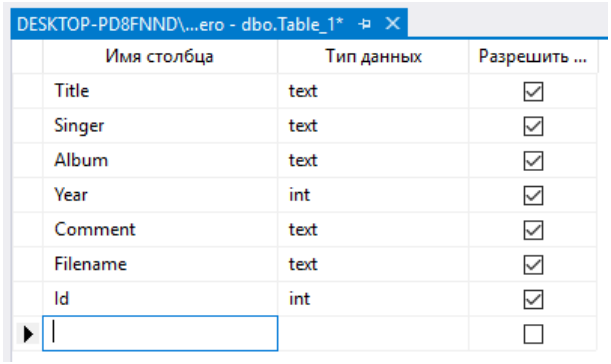
ПОДГОТОВКА БАЗЫ ДАННЫХ

Запишем прочитанную информацию а нашу базу данных **aero**. Для этого нам потребуется создать таблицу для их хранения. Откройте редактор **SQL Server Management Studio** и найдите в обозревателе объектов базу данных **aero**.

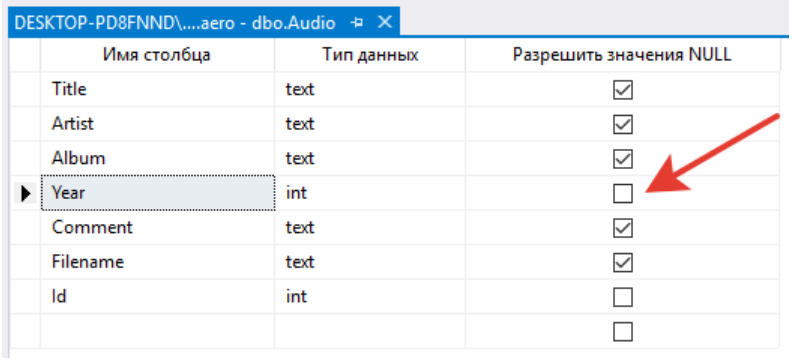
Щелкните правой кнопкой мыши по узлу Таблицы и выберите пункт выпадающего меню Таблица.



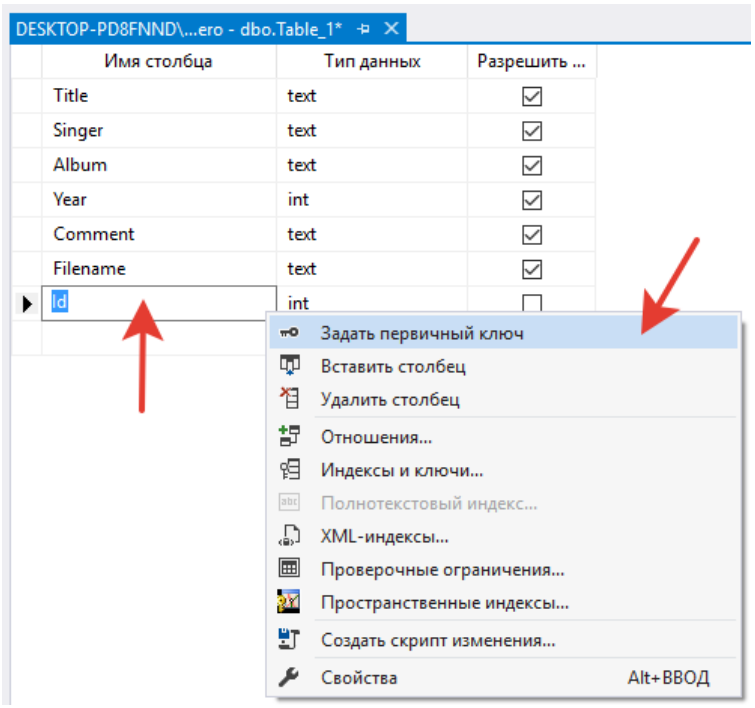
Откроется редактор таблиц. В нем мы будем настраивать параметры столбцов. Для добавления нового столбца просто начните вводить его имя в пустом поле. Создайте пять столбцов с типом **text** и именами: **Title, Singer, Album, Comment** и **Filename**. Затем создайте еще два столбца с типом **int** и именами **Year** и **Id**.



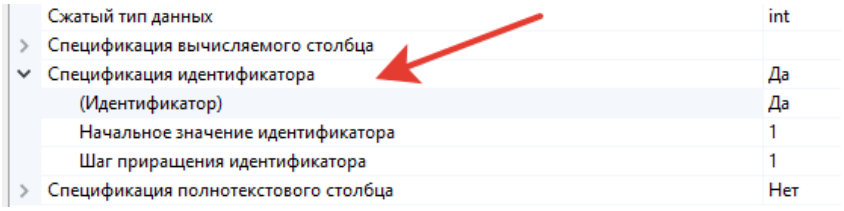
Уберите галочку в столбце «**Разрешить значения NULL**» для столбца **Year**.



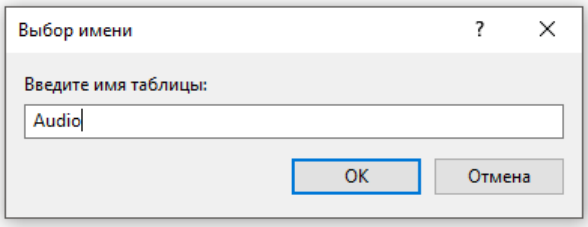
Нажмите правой кнопкой мыши на названии столбца **Id** и выберите пункт меню **Задать первичный ключ**.



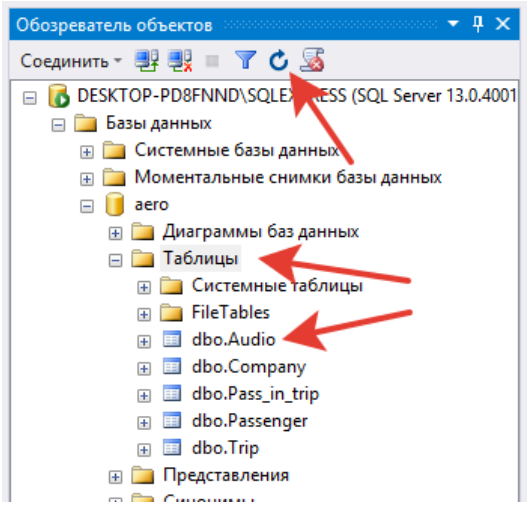
Нажмите правой кнопкой мыши по имени столбца **Id** и выберите пункт меню **Свойства**. В окне свойств найдите раздел **Спецификация идентификатора** и измените его содержимое как указано на рисунке:



Сохраните изменения нажав **Ctrl+S**. Вам будет предложено дать название для созданной таблицы. Введите имя **Audio**.



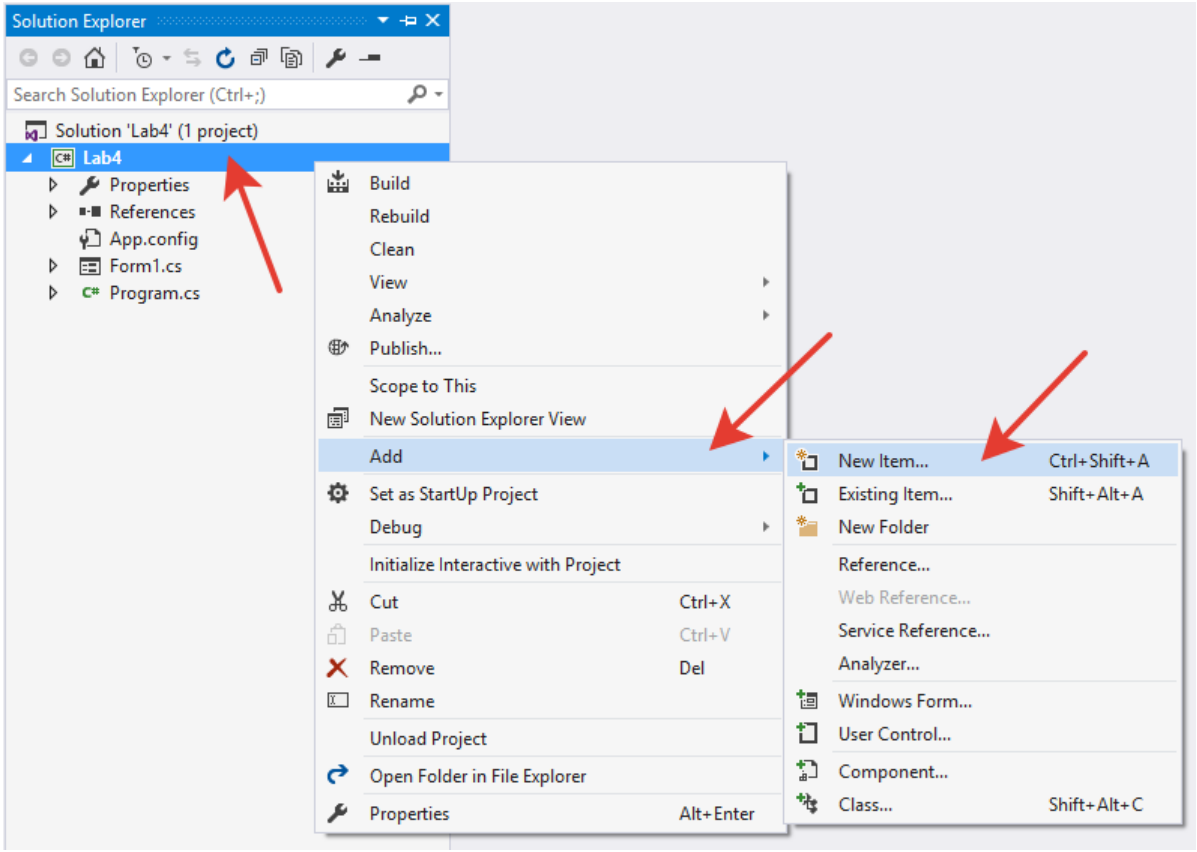
Изменения в базе данных могут сразу не отобразиться в обозревателе объектов. Вы можете выбрать узел **Таблицы** и нажать кнопку **Обновить** на панели инструментов. После этого в списке узлов должна появиться таблица **Audio**.



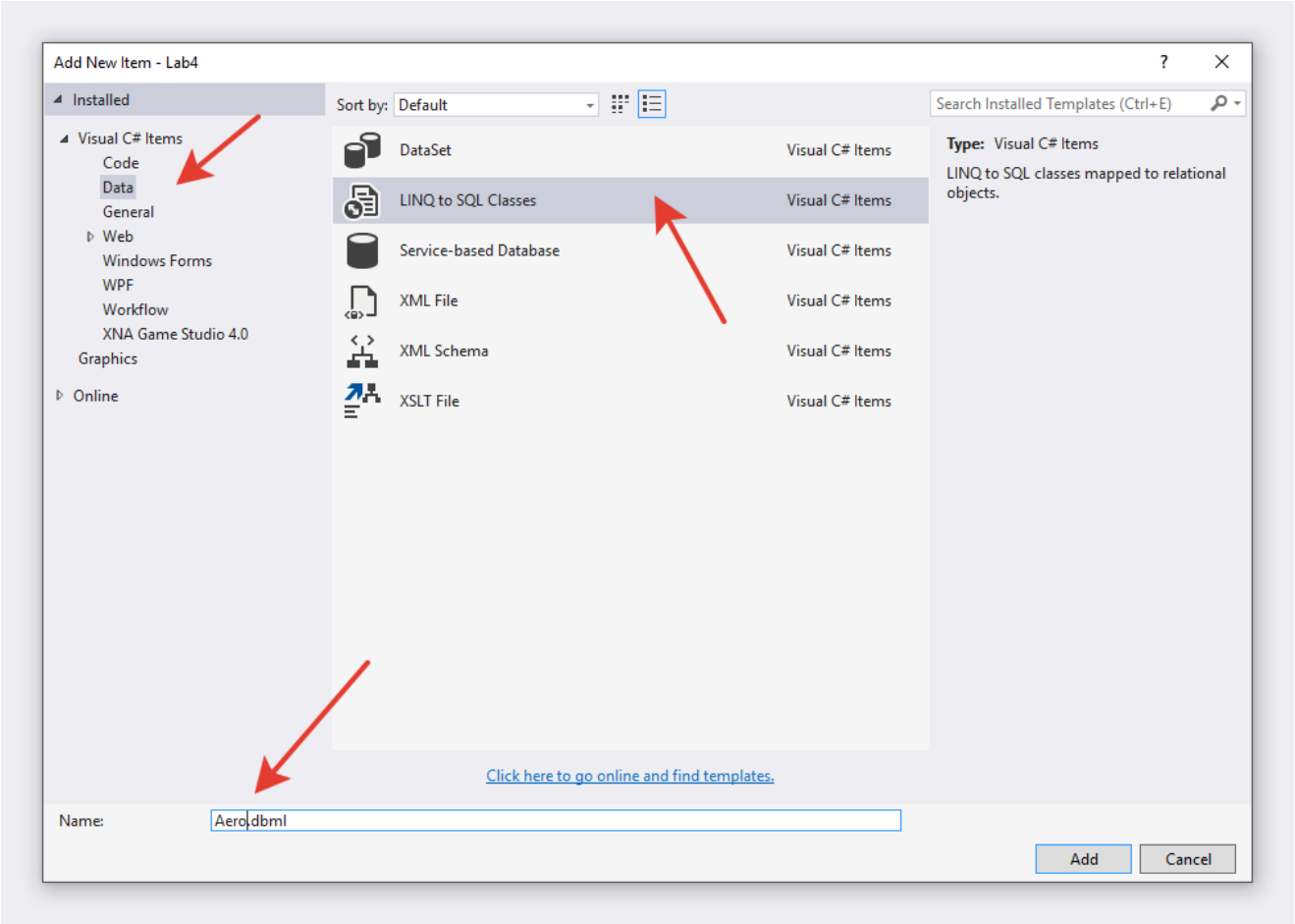
ОТПРАВКА ДАННЫХ В БАЗУ

Вернемся в редактор **Visual Studio**. Если вы выполняли лабораторную работу №3, то в **Обозревателе серверов** у вас должно было остаться подключение к нашей базе данных SQL. Если оно отсутствует, выполните первую часть лабораторной работы №3 и создайте его заново.

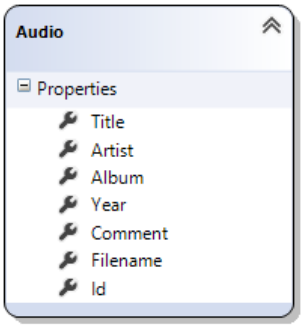
Создайте новый объект **LINQ to SQL**. Для этого нажмите правой кнопкой мыши на узле проекта в **обозревателе решений** и выберите пункты меню **Добавить – Новый элемент (Add – New Item)**.



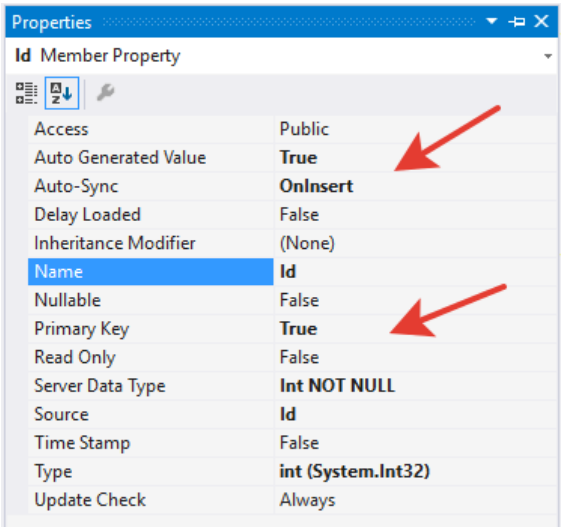
В открывшемся диалоговом окне найдите пункт **LINQ to SQL Classes** в разделе **Данные (Data)** для языка C#. В нижней части диалога введите имя **Aero**.



Откроется **конструктор отношений.** В **обозревателе серверов** (**Server Explorer**) откройте подключение к базе данных **aero** и найдите узел таблицы **Audio**. Перенесите узел таблицы в левую рабочую область, в конструкторе отношений появится графическое представление таблицы:



Нажмите правой кнопкой мыши на поле **Id** и выберите пункт меню **Свойства** (**Properties**). В панели свойств установите следующие параметры:



Теперь **класс Audio** в нашей программе будет автоматически обновлять первичный ключ **таблицы Audio**.

Вернемся к нашей форме. Откройте код кнопки **Открыть** (метод **open_Click**) и сразу после получения тегов аудио файла вставьте код отправки данных в базу SQL:

```
var db = new AeroDataContext();
var data = new Audio {
    Filename = filename,
    Title = file.Tag.Title,
    Artist = file.Tag.FirstPerformer,
    Album = file.Tag.Album,
    Year = (int)file.Tag.Year,
    Comment = file.Tag.Comment
};
db.Audios.InsertOnSubmit(data);
db.SubmitChanges();
```

Здесь мы сначала создаем объект базы данных:

```
var db = new AeroDataContext();
```

Затем мы создаем новую запись в таблице **Audio** и заполняем ее данными из тегов открытого аудио файла:

```
var data = new Audio {
    Filename = filename,
    Title = file.Tag.Title,
    Artist = file.Tag.FirstPerformer,
    Album = file.Tag.Album,
    Year = (int)file.Tag.Year,
    Comment = file.Tag.Comment
};
```

Регистрируем запись в очереди на обновление таблицы:

```
db.Audios.InsertOnSubmit(data);
```

И вызываем саму процедуру обновления:

```
db.SubmitChanges();
```

Теперь, при каждом открытии аудио файла в базу данных будет добавляться информация об этом файле:

Результаты Сообщения

	Title	Artist	Album	Year	Comment	Filename	Id
1	Homer's shout	The Simpsons	Samples	2017	This is a comment	C:\Users\igor\Desktop\Sample.wma	1
2	Homer's shout	The Simpsons	Samples	2017	This is a comment	C:\Users\igor\Desktop\Sample.wma	2

ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

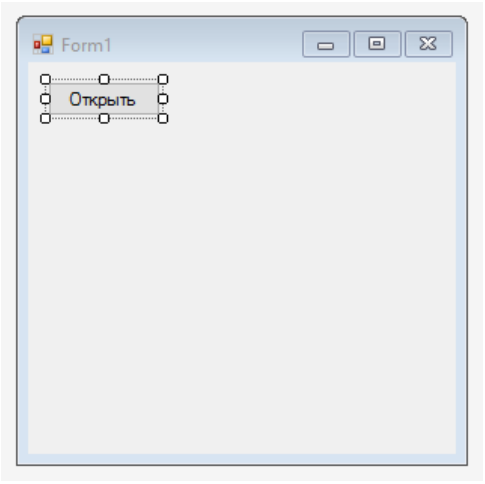
По аналогии с лабораторной работой №3 создайте программу, осуществляющую вывод содержимого таблицы **Audio** базы данных **aero**.

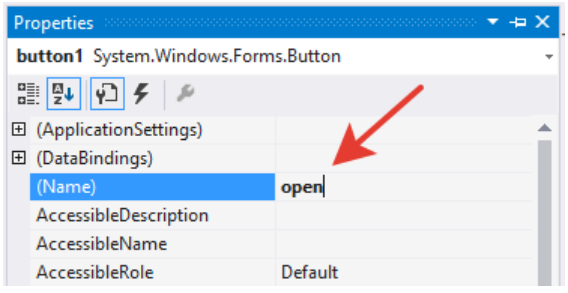
ЛАБОРАТОРНАЯ РАБОТА №2 (ЗАДАНИЕ 2)

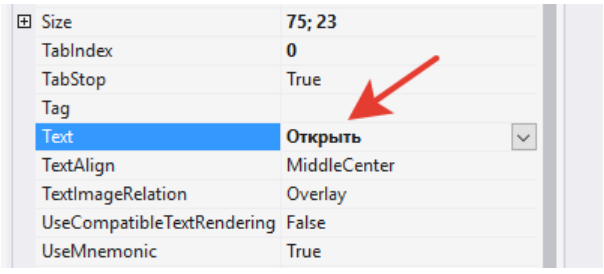
РАБОТА С ВИДЕО

Создадим новый проект Windows Forms. Выберите пункты меню **Файл – Новый – Проект** (**File – New – Project**). В открывшемся диалоговом окне выберите пункт **приложение Windows Forms** (**Windows Forms Application**). Введите Lab5 в качестве названия нового проекта.

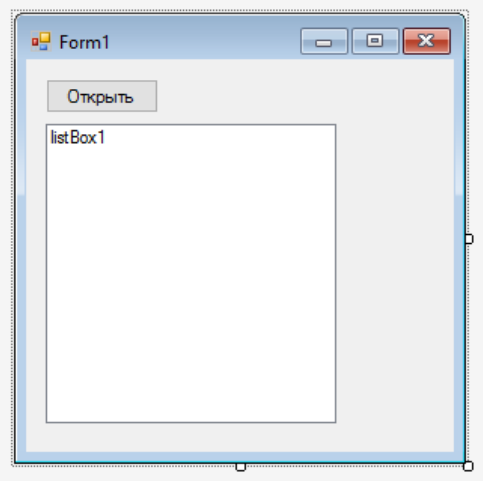
В редакторе форм добавьте на форму одну кнопку. В свойствах кнопки установите имя переменной (**Name**) равным «**open**», а текст кнопки измените на «**Открыть**».



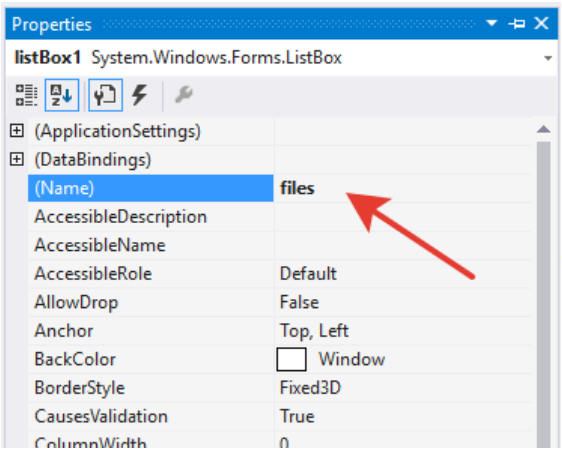




Добавьте на форму элемент управления **ListBox**. Этот элемент представляет собой список из произвольных объектов, мы будем использовать его для хранения очереди воспроизводимых файлов.



Откройте свойства элемента **ListBox** и измените название переменной списка с **listBox1** на **files**:



Добавьте обработчик нажатия на кнопку «**Открыть**» (двойной щелчок по кнопке в редакторе форм).

По аналогии с лабораторной работой №4 добавим всплывающее диалоговое окно открытия файла. Для этого создадим новый объект класса **OpenFileDialog** и вызовем метод **ShowDialog**:

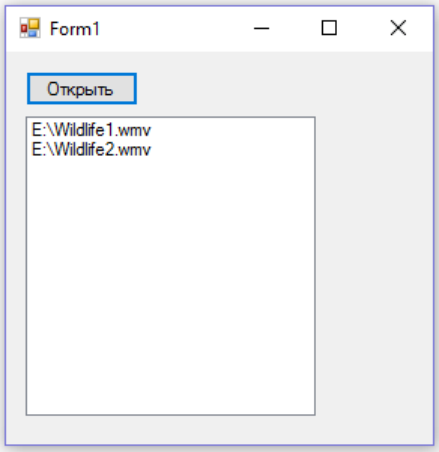
```
private void open_Click(object sender, EventArgs e) {
    var dialog = new OpenFileDialog();
    if (dialog.ShowDialog() == DialogResult.OK) {

    }
}
```

У элемента управления **ListBox** есть специальное свойство **Items**, представляющее собой список элементов типа **object**. Поэтому, мы можем поместить на хранение в **ListBox** объект любого класса, описанный в нашей программе. В списке он будет отображаться под значением, возвращаемым методом **ToString**.

Мы будем хранить в списке строки с полным маршрутом до видео файла. Получать путь будем из диалога:

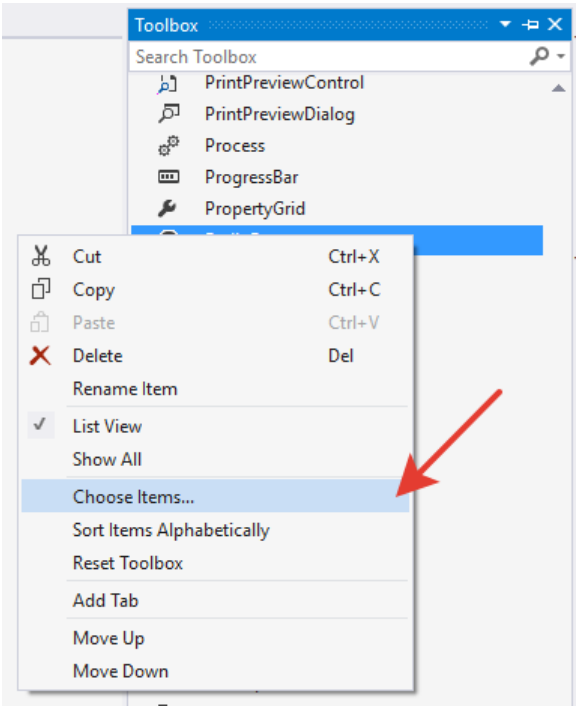
```
if (dialog.ShowDialog() == DialogResult.OK) {
    var filename = dialog.FileName;
    files.Items.Add(filename);
}
```

После запуска программы откроется форма приложения, при нажатии на кнопку «Открыть» можно выбрать файл и после этого полное имя файла отобразится в списке:

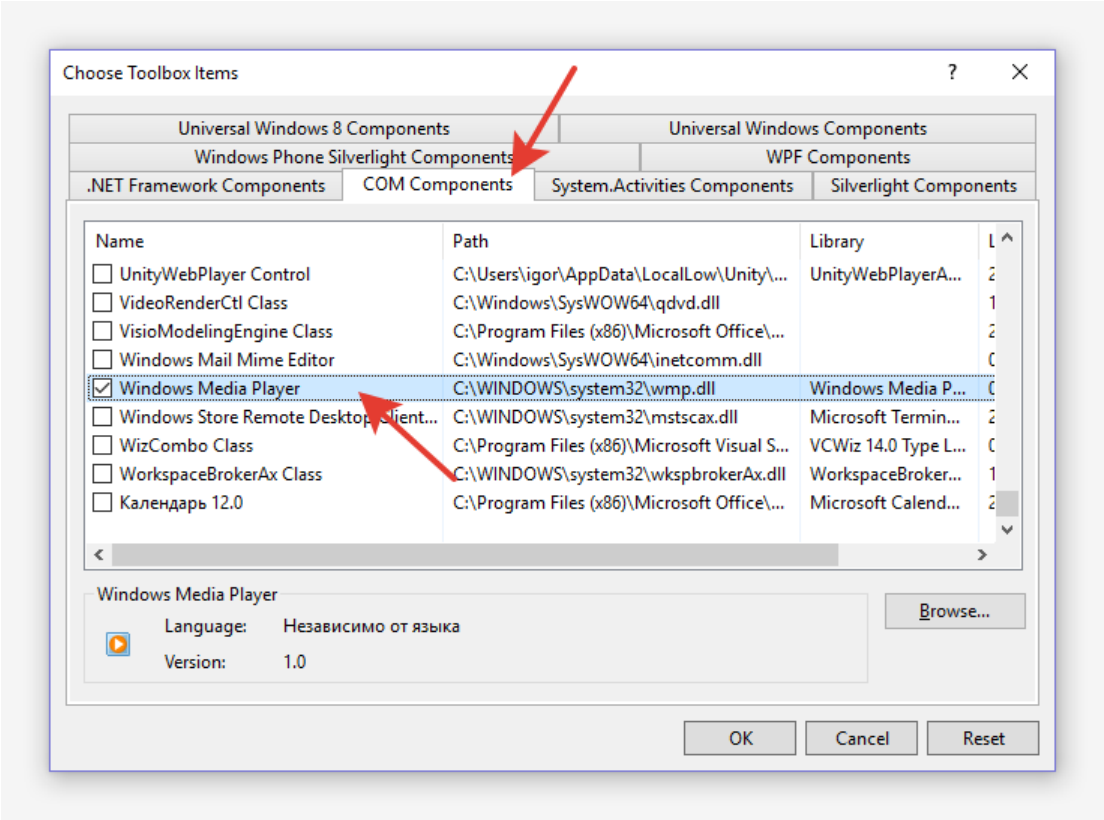


Добавим на форму компонент **Windows Media Player** для воспроизведения видео. По умолчанию этот компонент отсутствует на панели инструментов **Visual Studio**, поэтому нам нужно сначала его подключить.

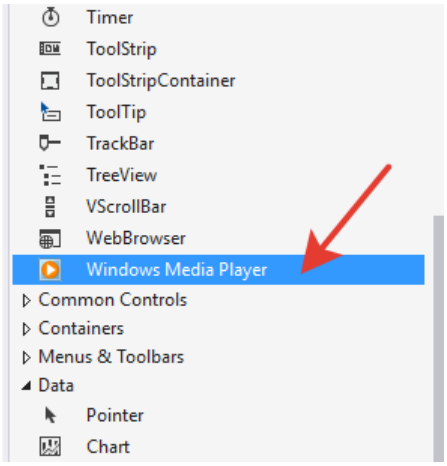
В редакторе форм откройте **панель инструментов** (**Toolbox**) и нажмите правой кнопкой мыши на любом ее элементе. Выберите пункт всплывающего меню **Выбор элементов (Choose Items)**.



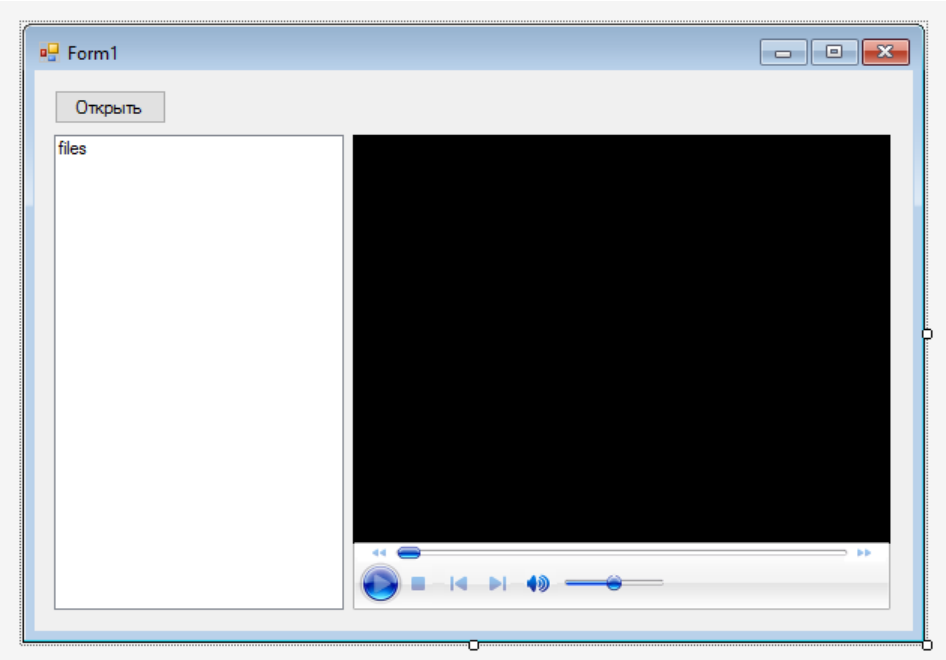
В открывшемся диалоговом окне выберите закладку **Компоненты COM (COM Components)**. В самом низу списка найдите компонент **Windows Media Player**. Выберите его и нажмите кнопку **ОК**.



В самом низу, на панели инструментов **Visual Studio** появится новый пункт:



Расширьте форму до удобного вам размера и перенесите на нее компонент проигрывателя:



Откройте свойства нового компонента и измените имя переменной (Name) с **axWindowsMediaPlayer1** на **player**.

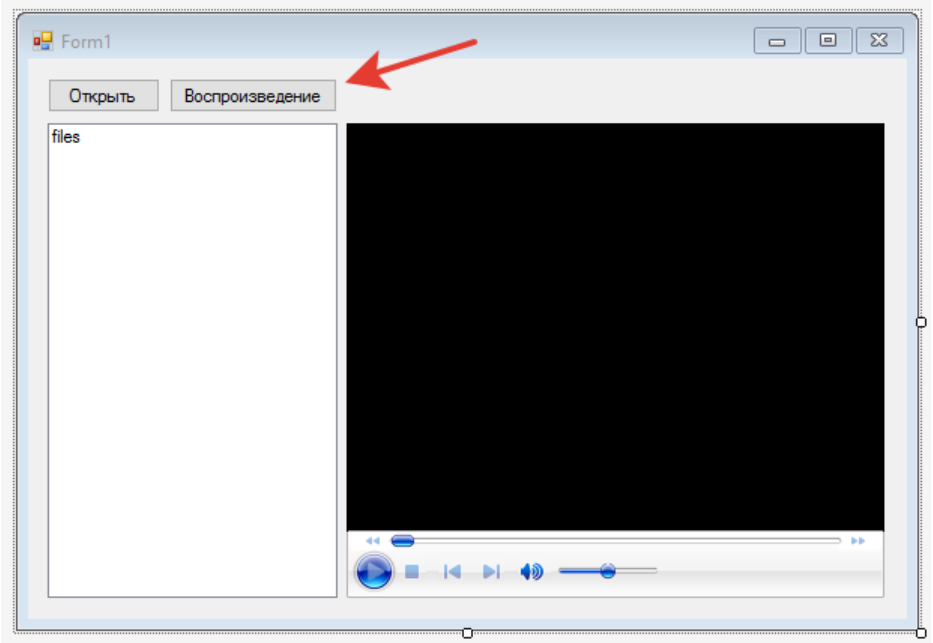
Пользоваться объектом плеера очень просто. Для воспроизведения файла достаточно присвоить свойству **URL** полный путь к файлу с видео:

```
player.URL = filename;
```

Если добавить этот код в метод **open_Click**, то воспроизведение начнется сразу после выбора файла на диске. Для прекращения воспроизведения достаточно вызвать метод **close**:

```
player.close();
```

Добавим возможность воспроизведения всех файлов из нашего списка **ListBox**. Для начала создадим на форме новую кнопку. В свойствах кнопки изменим имя переменной на **playAll**, а название кнопки на **Воспроизведение**.



Двойным щелчком по кнопке добавим обработчик события, он будет называться **playAll_Click**. В этом обработчике пропишем код для создания и заполнения списка воспроизведения. Сначала создадим сам объект плейлиста:

```
var playlist = player.newPlaylist("Playlist", "");
```

Для создания плейлиста мы вызываем метод **newPlaylist** у нашего объекта проигрывателя. Эта функция принимает два аргумента – название списка воспроизведения и путь к файлу со списком. Поскольку мы создаем абсолютно новый плейлист, то вместо пути просто передадим пустую строку.

Далее, в цикле заполним плейлист именами файлов из нашего компонента **ListBox**:

```
foreach (var selected in files.Items) {
    string s = selected.ToString();
    var temp = player.newMedia(s);
    playlist.appendItem(temp);
}
```

Имена файлов из списка **files.Items** хранятся в виде объектов типа **object**, поэтому перед использованием их необходимо преобразовать в строки. Делается это при помощи метода **ToString**:

```
string s = selected.ToString();
```

Воспроизведение начнется сразу после того, как мы присвоим наш новый плейлист свойству **currentPlayList** объекта **player**:

```
player.currentPlaylist = playlist;
```

ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

Используя инструкции из лабораторной работы №4, сделайте следующее:

1. Через свойства диалога **OpenFileDialog** ограничьте типы открываемых файлов расширением WMV.
2. Определите **ширину** и **высоту** исходного видео при помощи библиотеки **TagLib** (используйте свойства **Properties.VideoWidth** и **Properties.VideoHeight**).
3. В базе данных aero создайте новую таблицу Video. В таблице задайте следующие столбцы:
    a. Идентификатор **ID** – целое число, первичный ключ, автоматическое приращение.
    b. Имя файла **Filename** – строка символов (тип **Text**).
    c. Ширина видео **Width** – целое число.
    d. Высота видео **Height** – целое число.
4. Используя библиотеку **LINQ to SQL,** реализуйте добавление в базу данных **aero** информации о видео файле. (Рекомендуется добавить код подключения к БД в обработчик кнопки «**Открыть**»).