

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ

Заведующий методическим кабинетом

_____ Е.В.Фалей

« _____ » _____ 2021 г.

Специальность: 2-40 01 01 «Программное обеспечение информационных технологий»

Дисциплина: «Основы алгоритмизации и программирование»

Лабораторная работа № 29
Инструкционно-технологическая карта

Тема: Создание и обработка главного и контекстного меню. Отображение последовательностей, массивов с помощью компонентов работы со списками и таблицами

Цель: Научиться создавать главное меню и подпункты контекстного меню в нем; вводить и выводить последовательности и массивы в оконное приложение; работать со списками различных данных и объектов; использовать COM-компоненты

Время выполнения: 2 часа

1. Краткие теоретические сведения. Пример выполнения программы

Главное меню – горизонтальная лента пунктов меню, каждая вкладка которых может раскрываться списком вложенных вкладок. Вкладки между собой группируются по принципу логической связи или совместного использования.

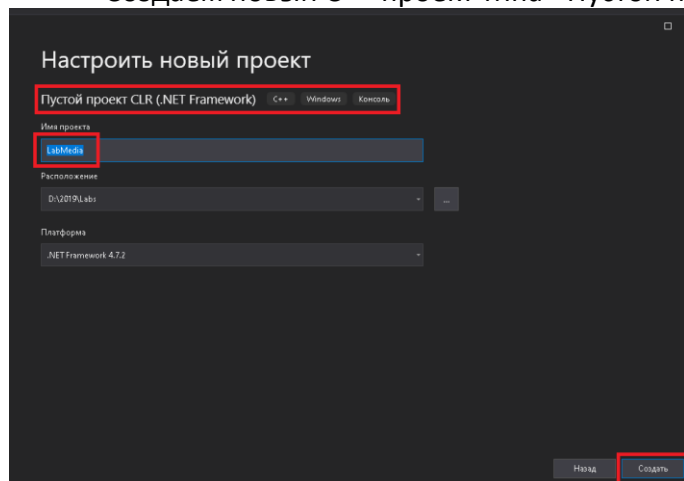
Контекстное меню – меню, не отображаемое на оконной форме, а доступное для вызова при нажатии на графический объект правой кнопкой мыши.

Списковый графический элемент для оконного приложения – визуальный компонент, который предназначен для отображения в виде вертикального (реже горизонтального) списка объектов из некоторого массива однотипных объектов. Если отдельные строки списка состоят из двух или более ячеек, то их можно считать визуальными таблицами, например, DataGridView.

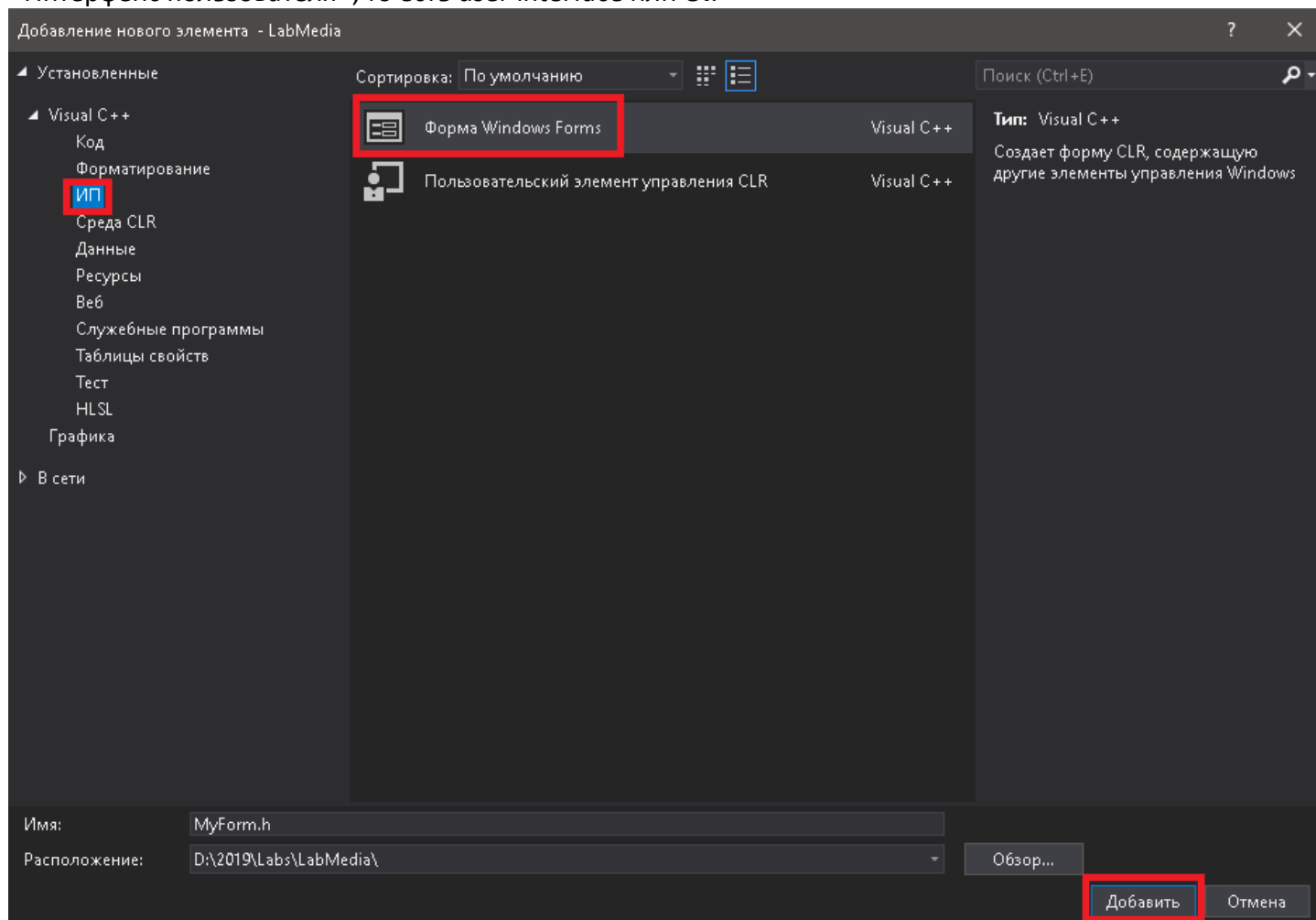
Компонент – COM-компонент – модульный скомпилированный бинарный исполнимый код, который имеет интерфейс взаимодействия с собой, зарегистрирован в главном каталоге операционной системы и может быть подключен к различным приложениям и работать в них как составная часть. COM-технология получила широкое распространение в ОС MS Windows.

Задание 1 (выполнить)

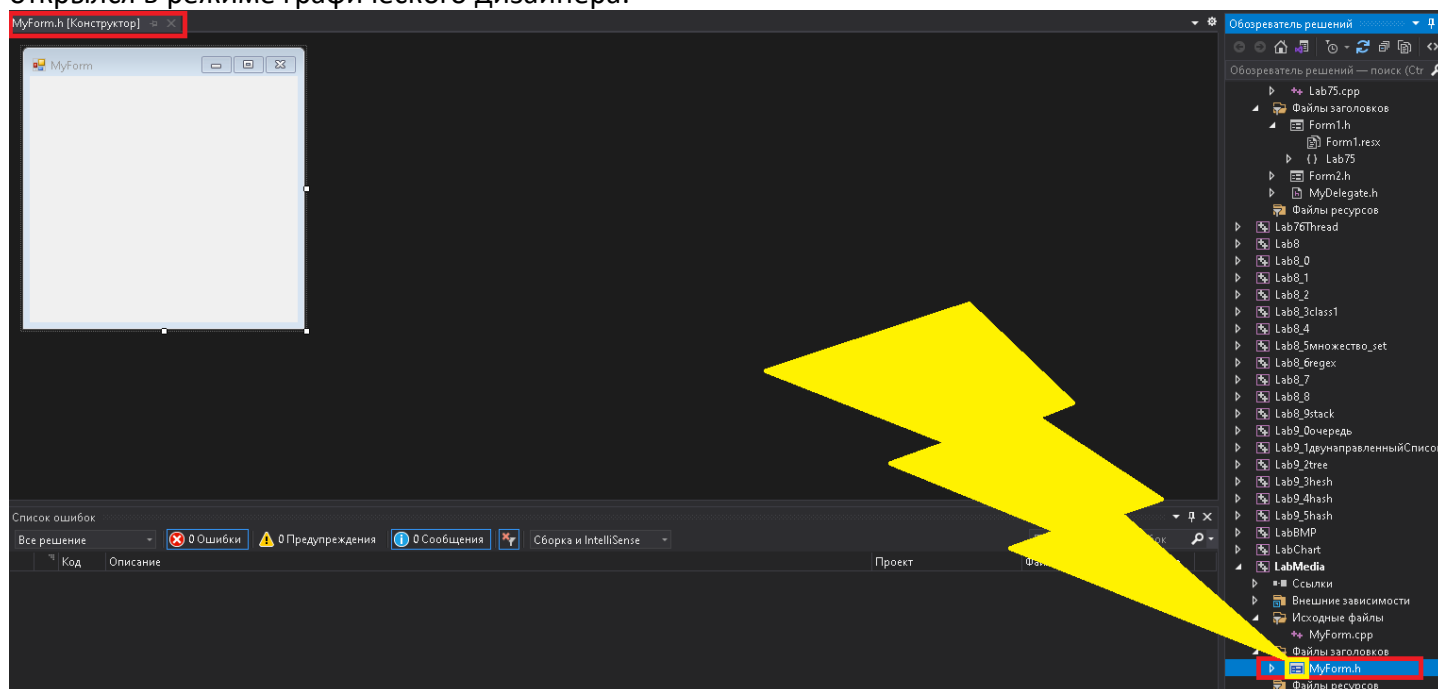
Создаем новый C++ проект типа «Пустой проект CLR (.NET Framework)» с названием LabMedia.



В проекте создаем главную оконную форму, нажимая правой кнопкой мыши по проекту / Добавить / Создать элемент / Visual C++ / ИП / Форма Windows Forms / Добавить. **ИП** – это аббревиатура от «Интерфейс пользователя», то есть user interface или **UI**.



Поскольку файл окна MyForm.h в режиме визуального дизайнера («Конструктора») не открылся сразу, то вкладки были закрыты, проект сохранен и MS Visual Studio перезагружена (выключена и включена повторно), после чего иконка (пиктограмма) хедер-файла MyForm.h стала в виде «окошка» и файл открылся в режиме графического дизайнера:

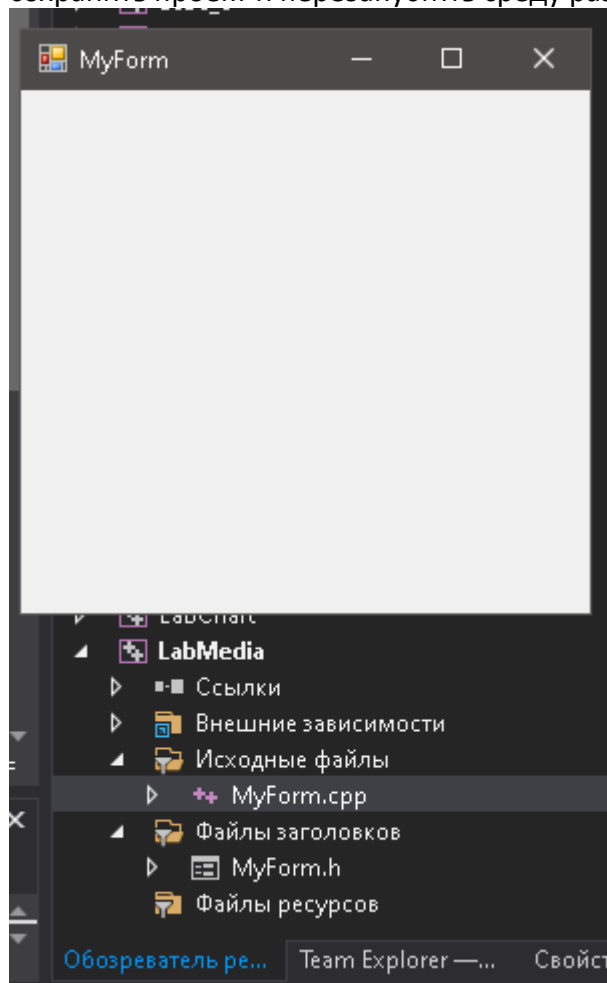


Открываем и дописываем MyForm.cpp файл, помня, что в каждом проекте название пространства имен совпадает с названием этого проекта, в нашем случае “LabMedia”.

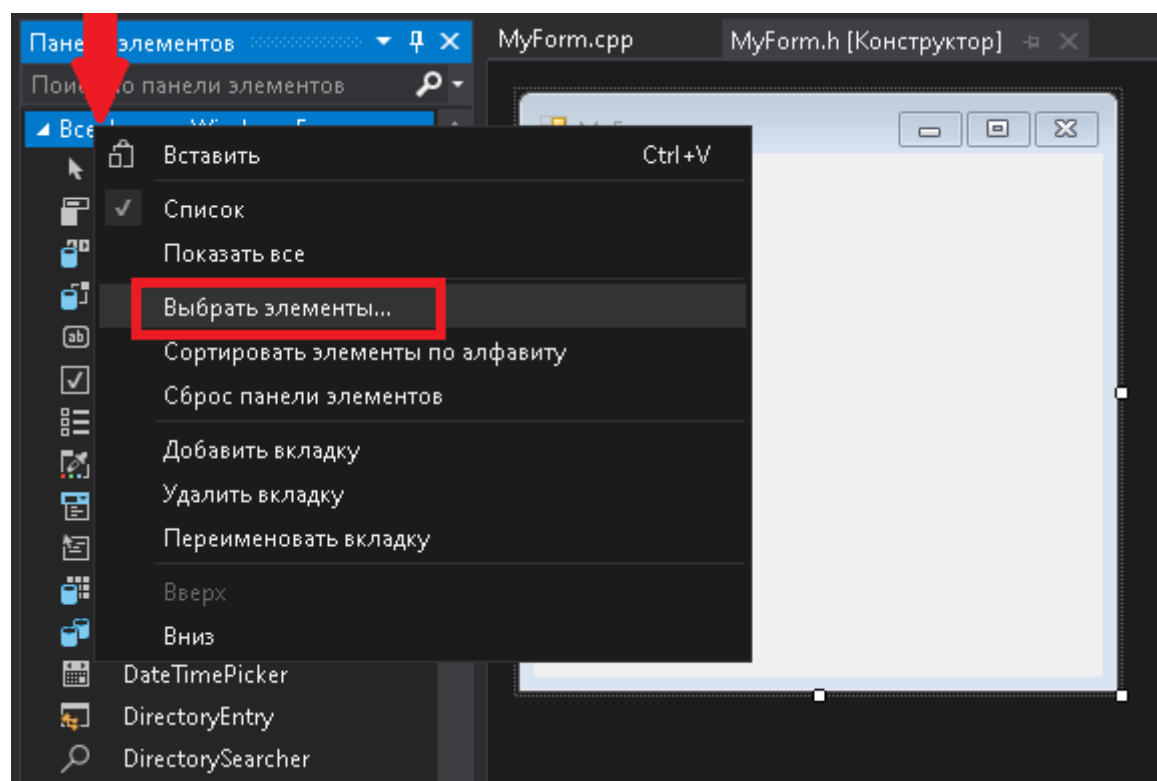
```
MyForm.cpp  x  MyForm.h [Конструктор]
LabMedia    (Глобальная область)  WinMain(HINSTANCE, HINSTANCE, LPSTR, int)

1  #include "MyForm.h"
2  #include <Windows.h>
3  using namespace LabMedia;
4
5  [STAThreadAttribute]
6  int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
7  {
8      Application::EnableVisualStyles();
9      Application::SetCompatibleTextRenderingDefault(false);
10     Application::Run(gcnew MyForm()); //запускаем приложение, в котором создаем экземпляр окна MyForm с помощью конструктора без параметров MyForm()
11     return 0;
12 }
```

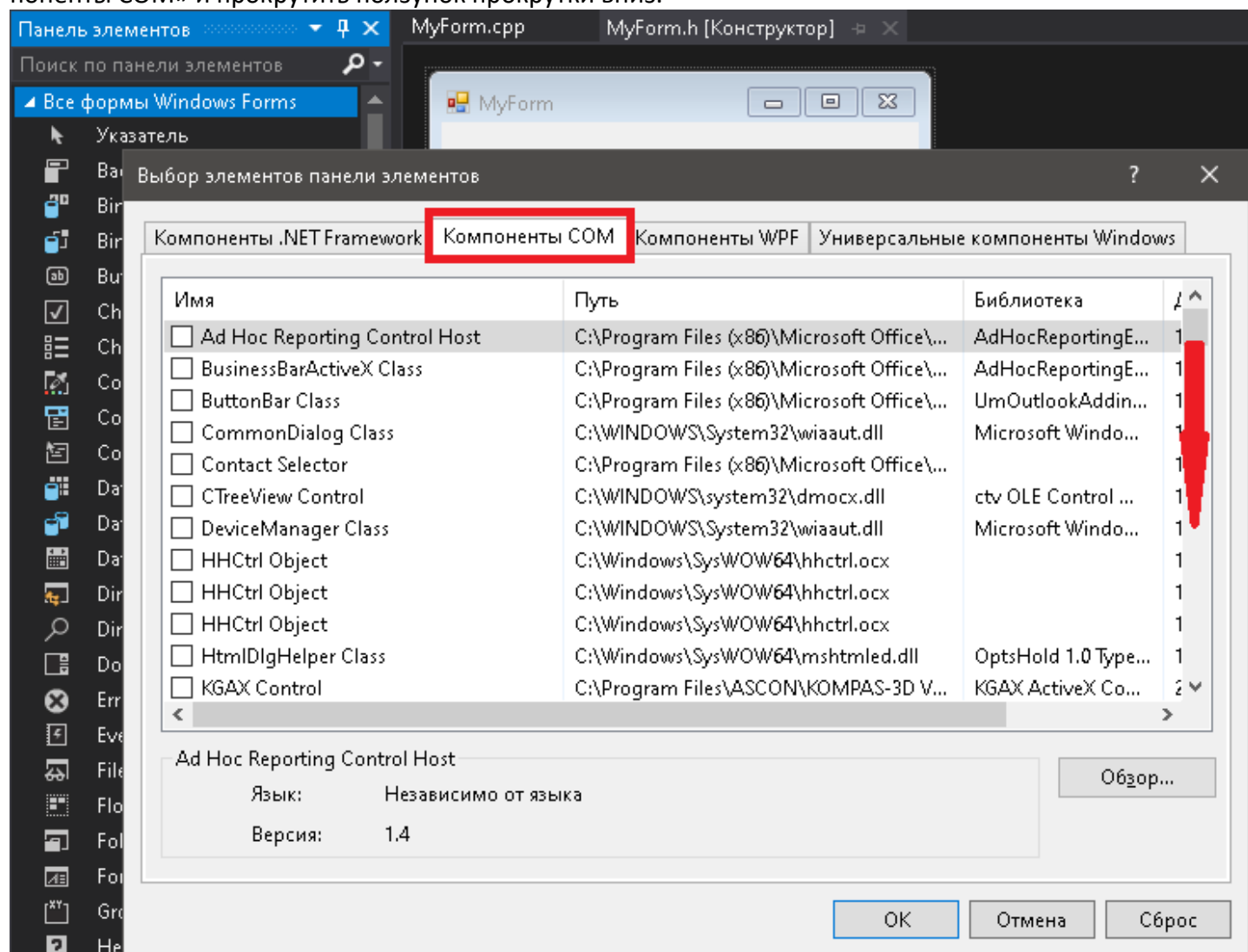
Проверяем, что проект назначен в качестве автозапускаемого, сохраняем его, компилируем и запускаем проект. Если нет ошибок, то отработает оконная форма вашего нового приложения с базовым функционалом. В ином случае надо перепроверить код, а также, возможно, может потребоваться сохранить проект и перезапустить среду разработки MS VS 2019.



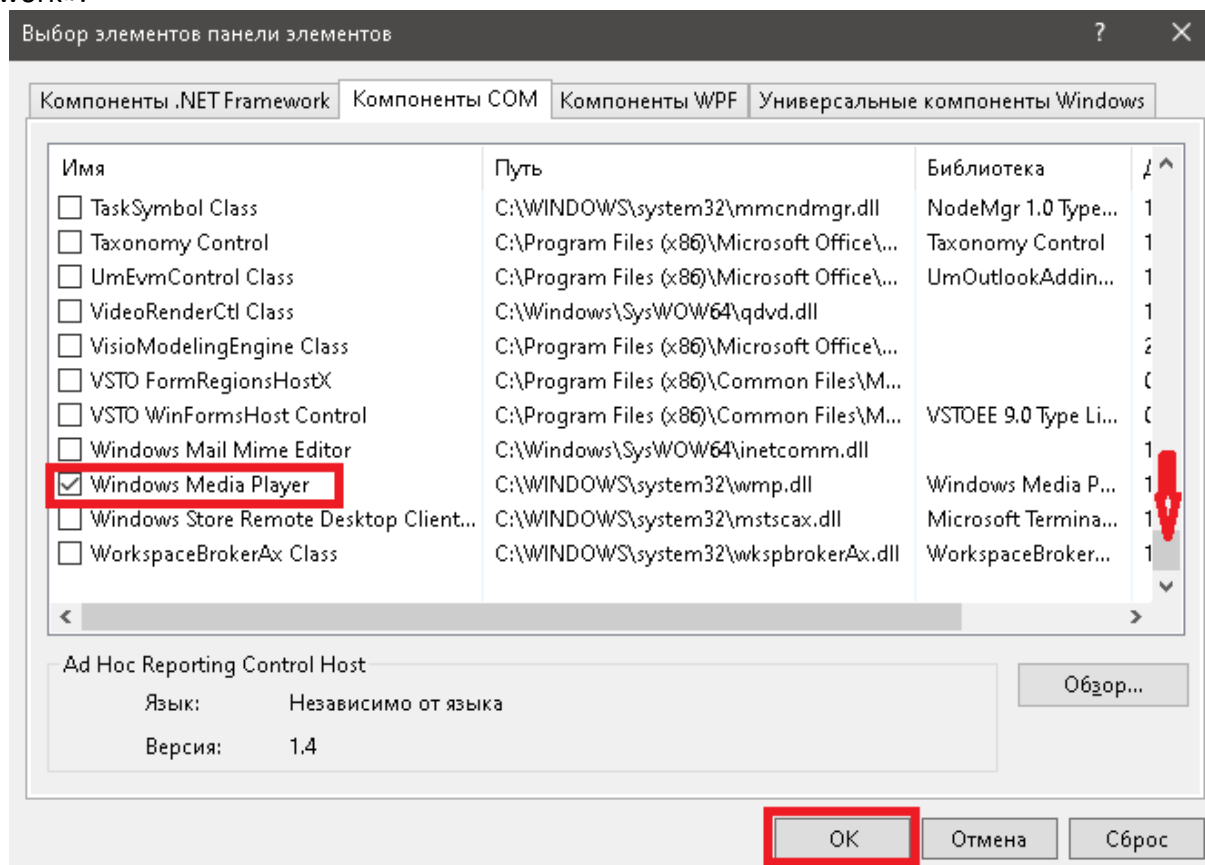
Во вкладке графического дизайнера нажать правой кнопкой мыши по подвкладке «Все Windows Forms» и выбрать пункт контекстного меню «Выбрать элементы».



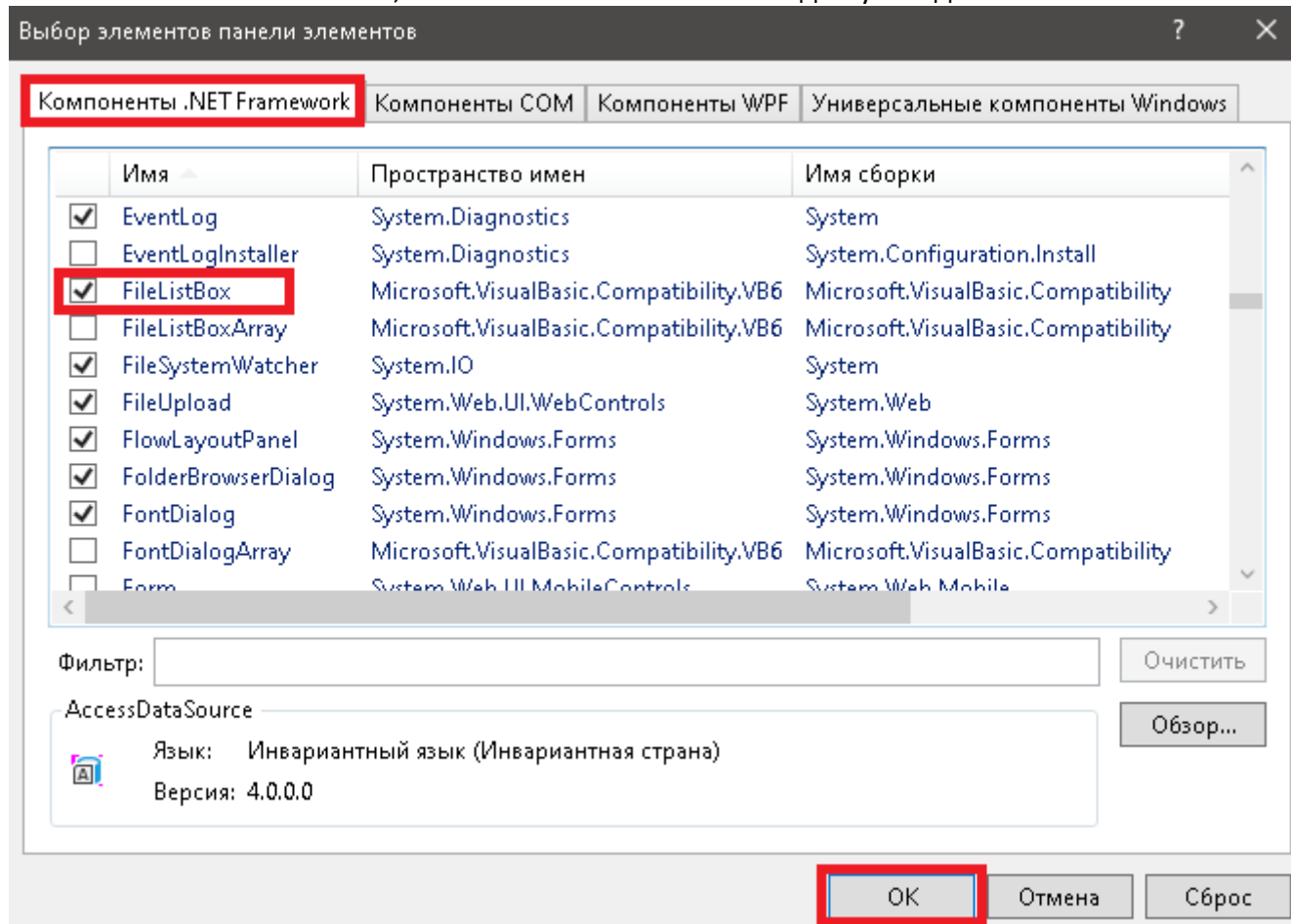
Дождаться загрузки окна «Выбор элементов панели элементов» и в нем выбрать вкладку «Компоненты COM» и прокрутить ползунок прокрутки вниз.



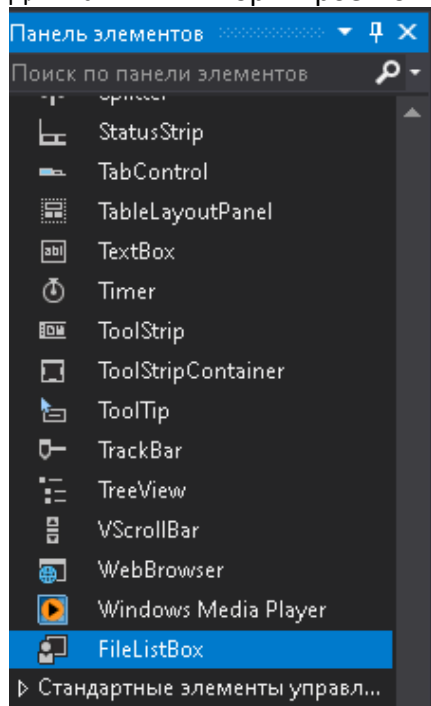
Поставить галочку в чекбоксе «Windows Media Player» и открыть вкладку «Компоненты .NET Framework».



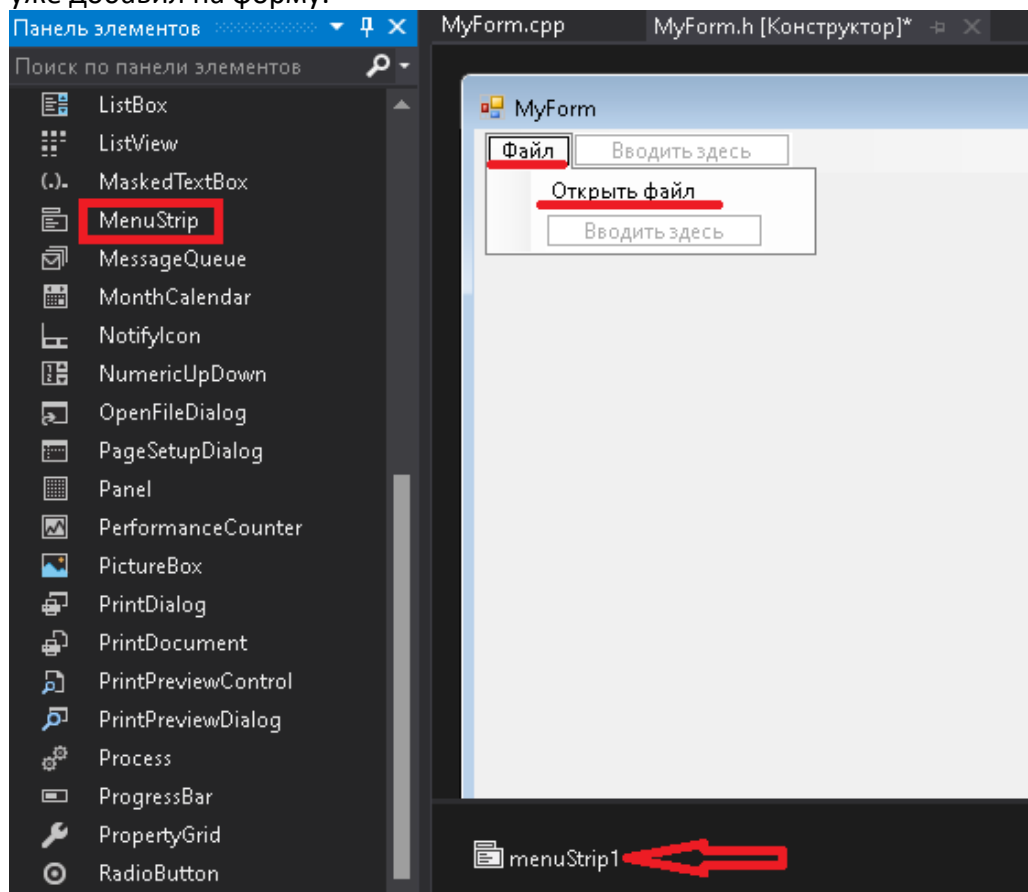
На вкладке «Компоненты .NET Framework» ставим галочку в чекбокс напротив компонента «FileListBox» и нажимаем ОК, чтобы эти компоненты стали доступны для использования в ВинФормах.



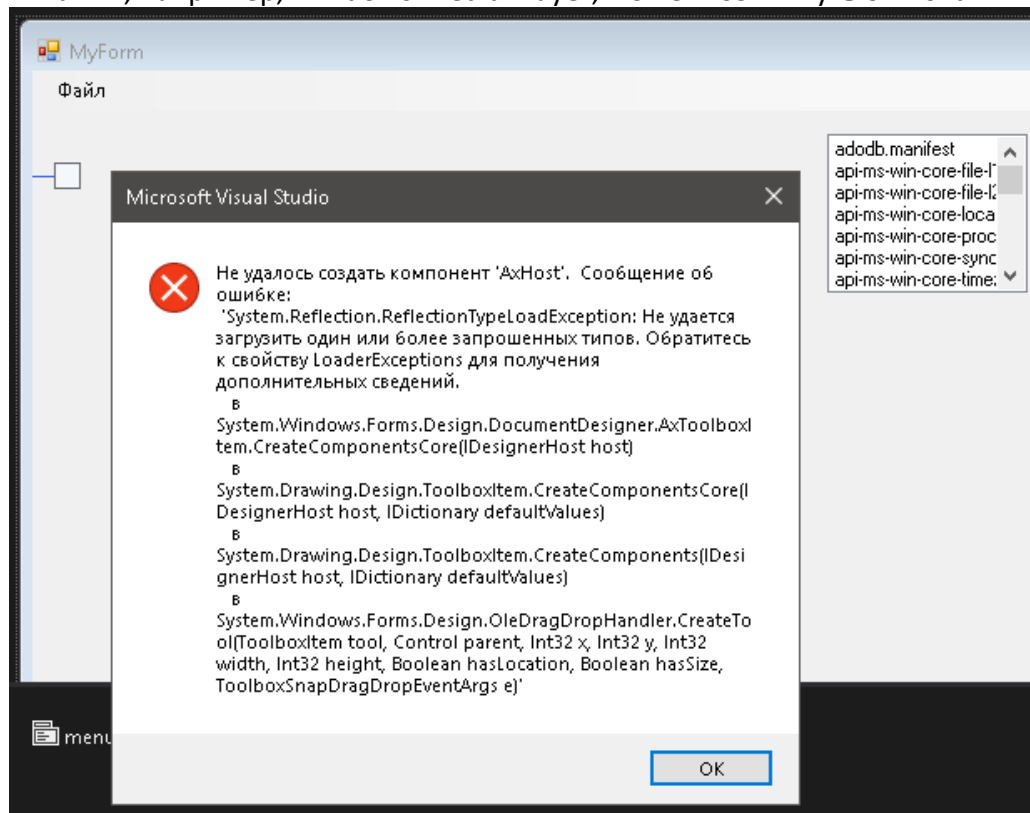
На вкладке «Все Windows Forms» ищем добавленные компоненты «Windows Media Player» и «FileListBox». Если они стали видны – готово. Разумеется, сам МедиаПлеер должен быть установлен как программа в ОС MS Windows 10 на вашем компьютере, и в этом случае он доступен как COM-компонент для ваших ВинФорм-проектов после описанных манипуляций.



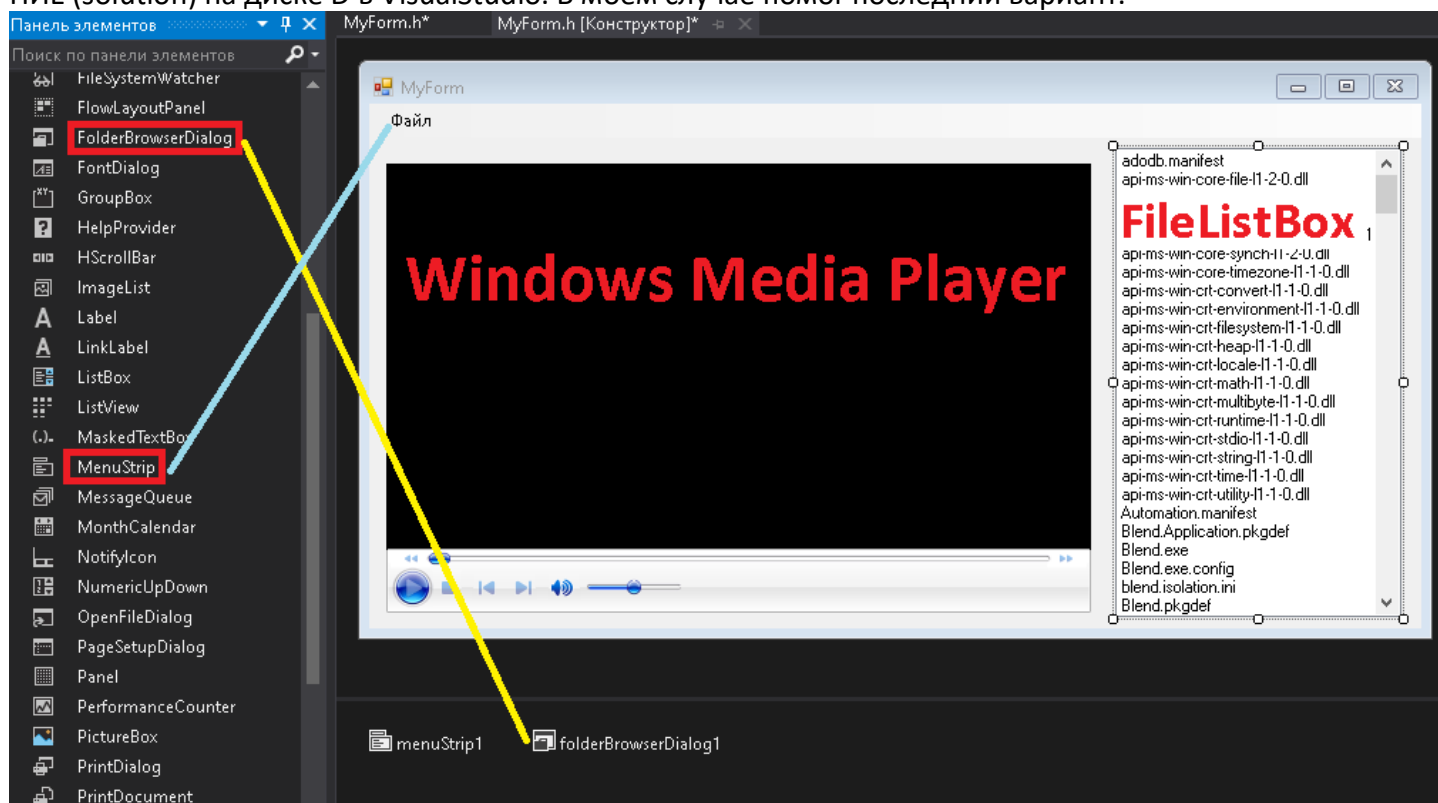
В режиме графического конструктора из Панели элементов перетащим мышью на оконную форму компонент MenuStrip. Это элемент «Главное меню», в котором можно делать Вкладки и Подвкладки в них. Поставьте курсор и наберите название вкладки «Файл» и под ней подвкладки «Открыть файл». Внизу дизайнера (под макетом оконной формы) должна отобразиться пиктограмма компонента menuStrip1 (это автосгенерированное имя объекта), поскольку меню может быть невидимым, а невидимые компоненты, перетянутые на окно, отображаются внизу окна, чтобы разработчик видел, что именно он уже добавил на форму.



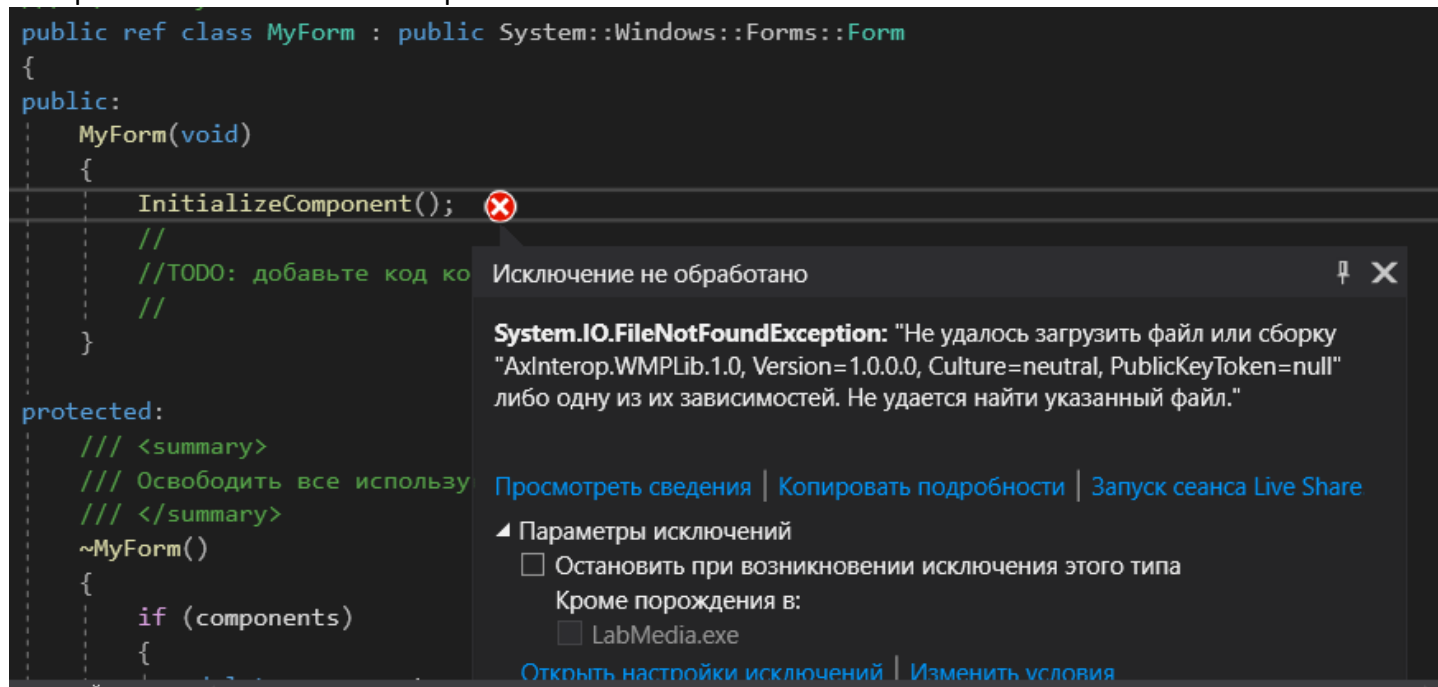
Также на макет оконной формы надо перетянуть мышью FolderBrowserDialog (скрытый компонент – он отобразится в программе, только если его вызовет пользователь своим действием), FileListBox, Windows Media Player (поместить и растянуть за ограничительные квадратики). При перетаскивании, например, Windows Media Player, может возникнуть ошибка:



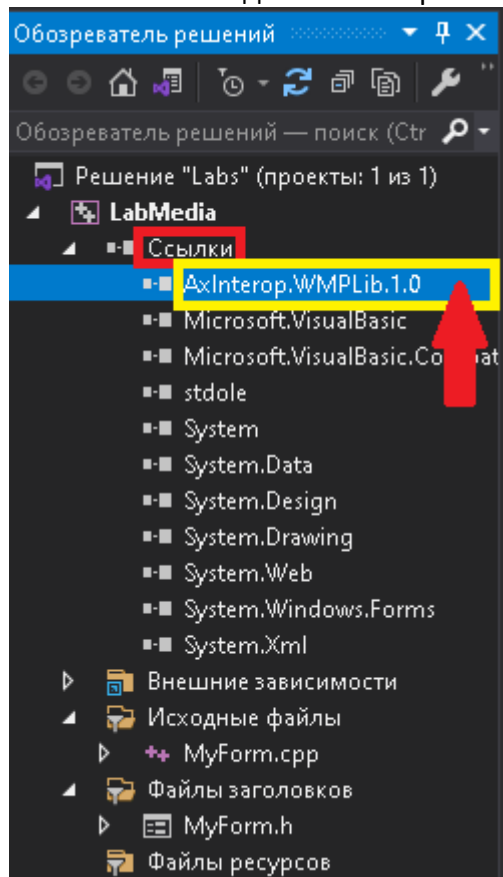
Эта ошибка возникла в связи с тем, что решение и проект были созданы, когда Windows Media Player еще не был подключен в качестве COM-компонента, поэтому, несмотря на его правильное добавление в Панель элементов, он автоматически не подключается на лету в существующий проект. Для исправления этой ошибки может понадобиться: 1) сохранить проект и перезапустить VisualStudio; 2) создать новый проект; 3) перезагрузить компьютер и 4) перезагрузить компьютер и создать новое РЕШЕНИЕ (solution) на диске D в VisualStudio. В моем случае помог последний вариант.

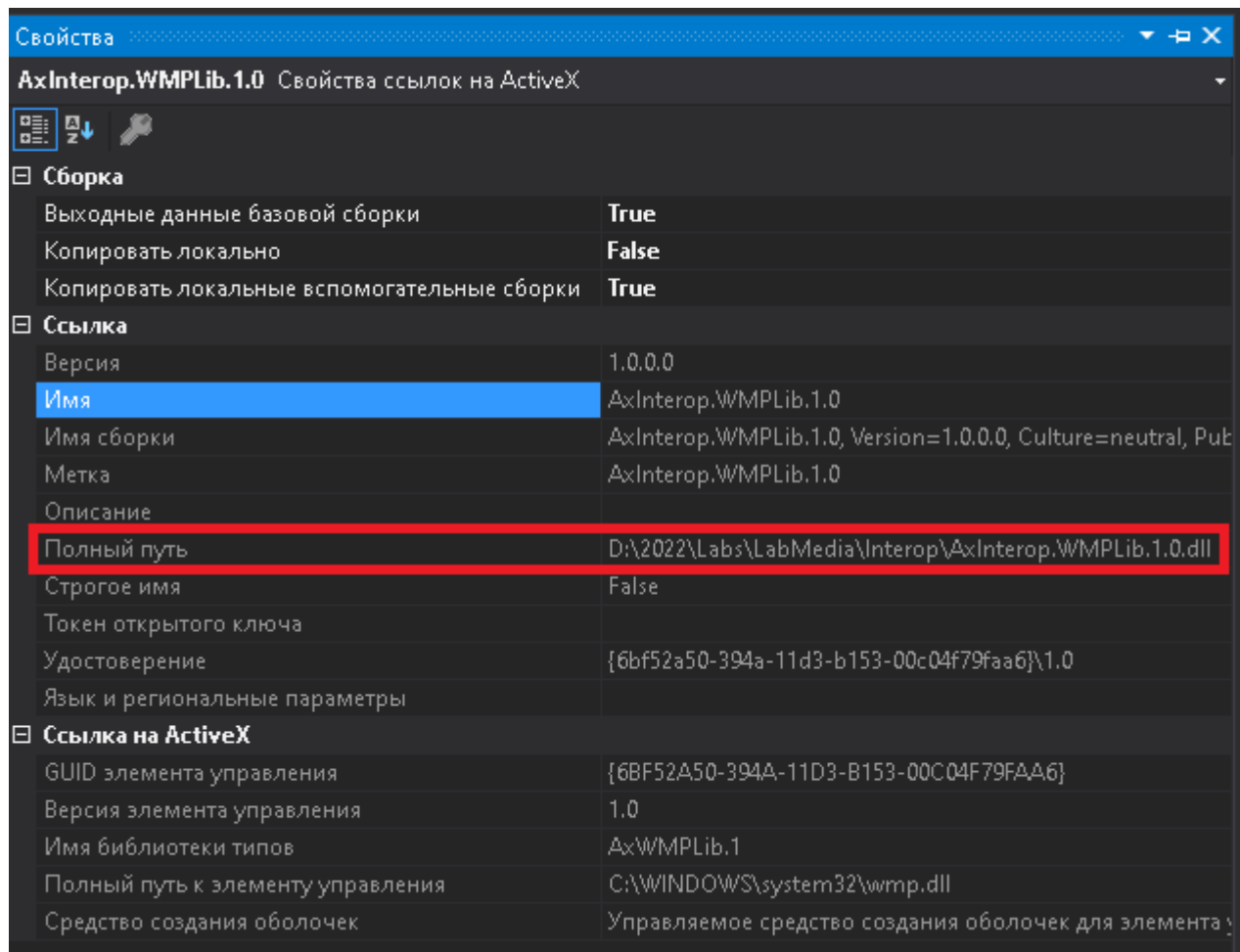


Если все 4 компонента добавлены и WinMain() с телом написан, можно компилировать проект. В моем случае выдало ошибку (см. ниже), то есть приложение LabMedia.exe не обнаружило рядом с собой файла-библиотеки AxInterop.WMPLib.dll

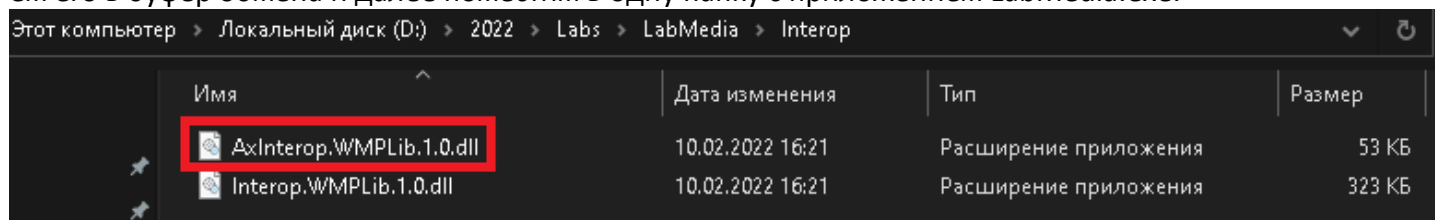


Нужно найти этот файл и поместить в папку вместе с приложением LabMedia.exe. Для этого в Обозревателе решений откроем вкладку Ссылки и вызовем контекстное меню у AxInterop.WMPLib.1.0.dll с помощью правой кнопки мыши и выбора пункта Свойства, чтобы узнать фактическое местонахождение этого файла.

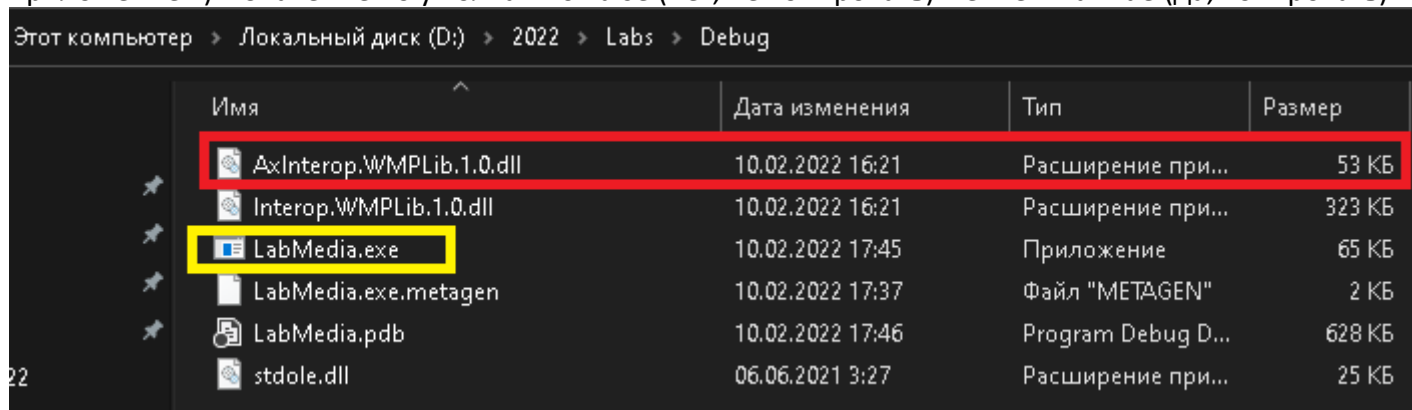




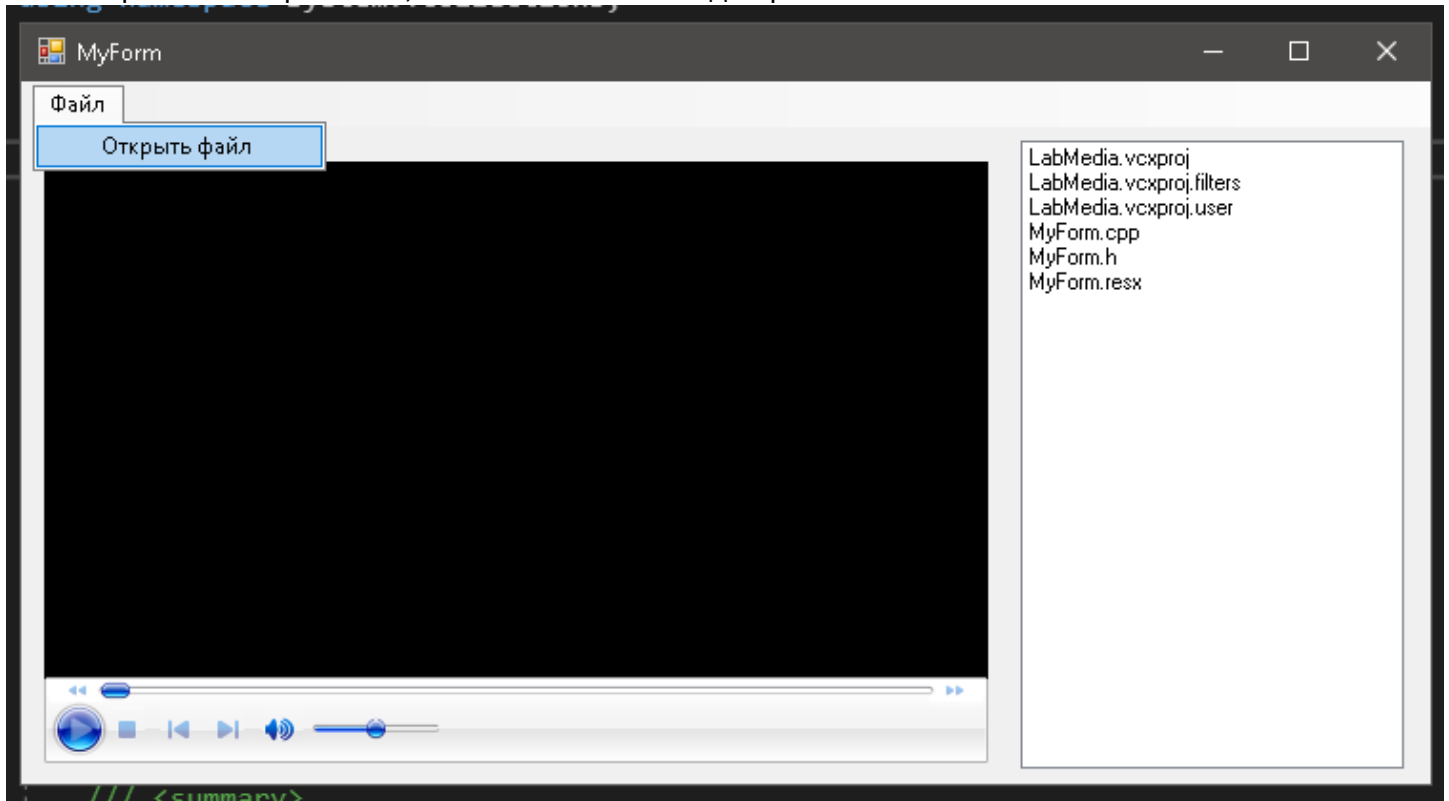
Идем в Проводник файлов и папок, проходим по данному пути и находим данный файл, копируем его в буфер обмена и далее поместим в одну папку с приложением LabMedia.exe.



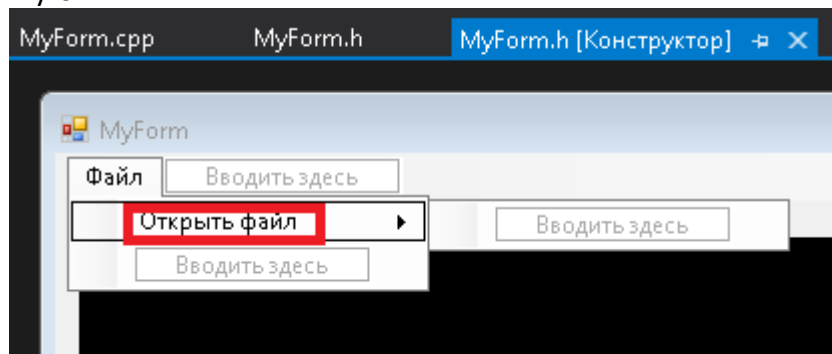
Само приложение LabMedia.exe обычно находится в папке Debug (если у вас режим Отладки, или Release – если у вас режим Релиза (Готовый, Финальный)) выше по дереву папок проекта, но в текущем решении solution. Если открыть эту папку, то в ней будет LabMedia.exe, но не будет файла AxInterop.WMPLib.1.0.dll, который мы сюда копируем из буфера обмена и перекомпилируем проект. Еще вариант решения проблемы: в окне Свойства (см. выше) найденного файла AxInterop.WMPLib.1.0.dll в разделе Сборка находим пункт Копировать локально (то есть в локальный каталог, местную папку с приложением) и значение по умолчанию False (нет, не копировать) меняем на True (да, копировать).



Приложение работает, осталось написать код обработчиков событий:

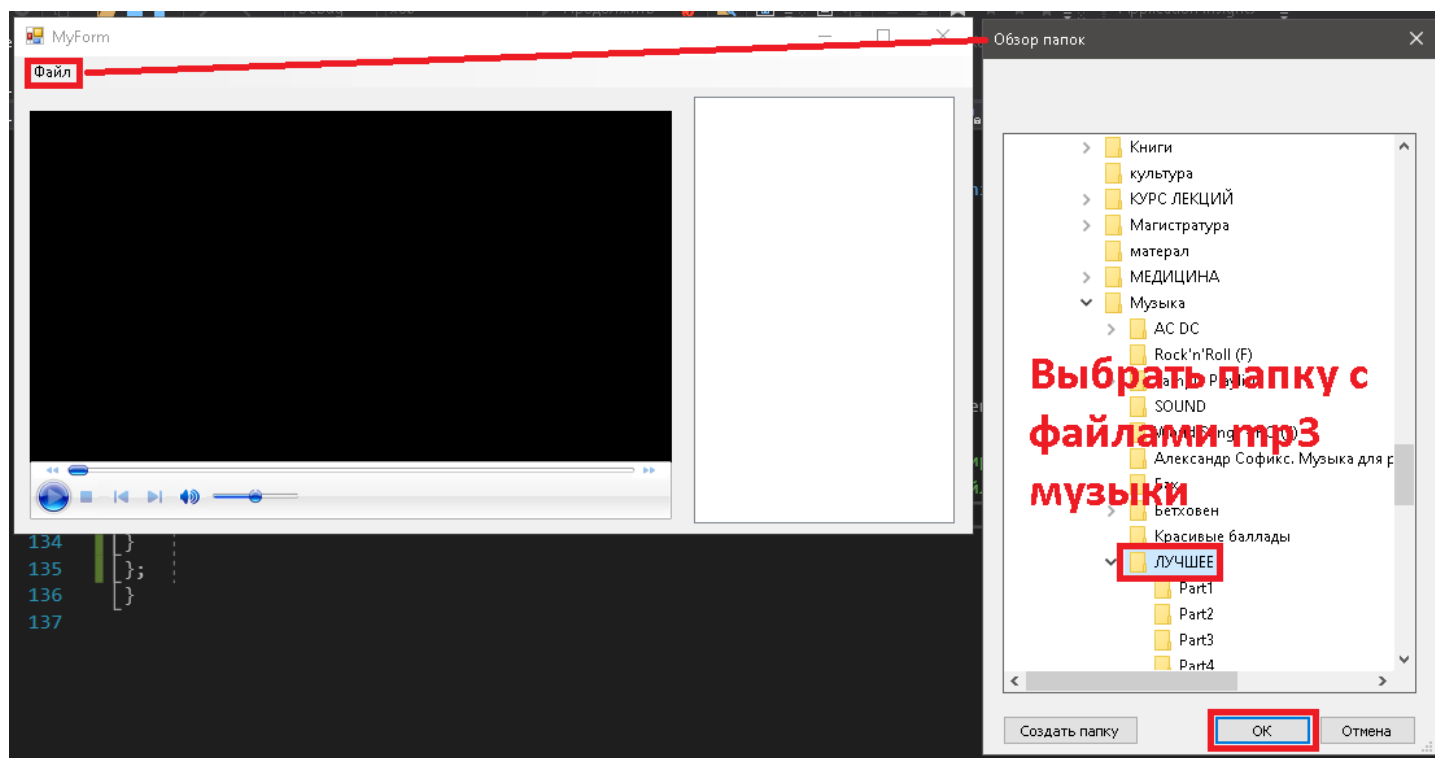


Для создания обработчика события нажатия пользователем на пункт меню «Открыть файл»: в режиме Конструктора мышью дважды кликаем по подпункту меню «Открыть файл» и переходим в код файла MyForm.h:

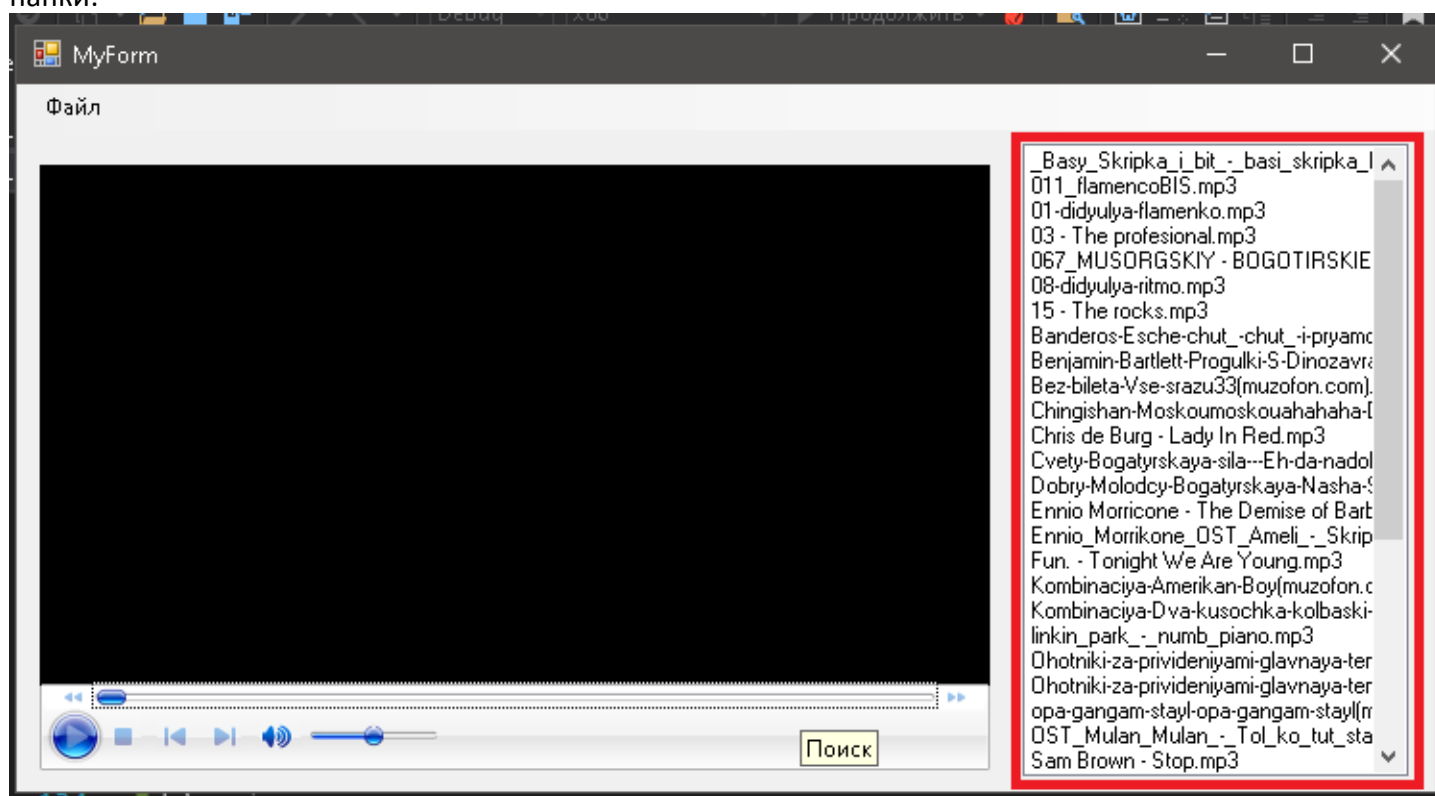


```
128 #pragma endregion
129 private: System::Void открытьФайлToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) //обработчик клика мыши по пункту меню "Открыть файл"
130 {
131     fileListBox1->Pattern = "*.mp3"; //оставим в списке только музыкальные файлы с расширением mp3, например
132     folderBrowserDialog1->ShowDialog(); //открыть окно просмотра папок и файлов как обязательное (модальное)
133     fileListBox1->FileName = folderBrowserDialog1->SelectedPath; //в список файлов отобразить только файлы из папки, путь к которой выбрали в окне диалога
134     // выбора папок, причем отобразит только mp3-файлы, а если их в выбранной папке не окажется - список будет пустой
135 };
136 }
```

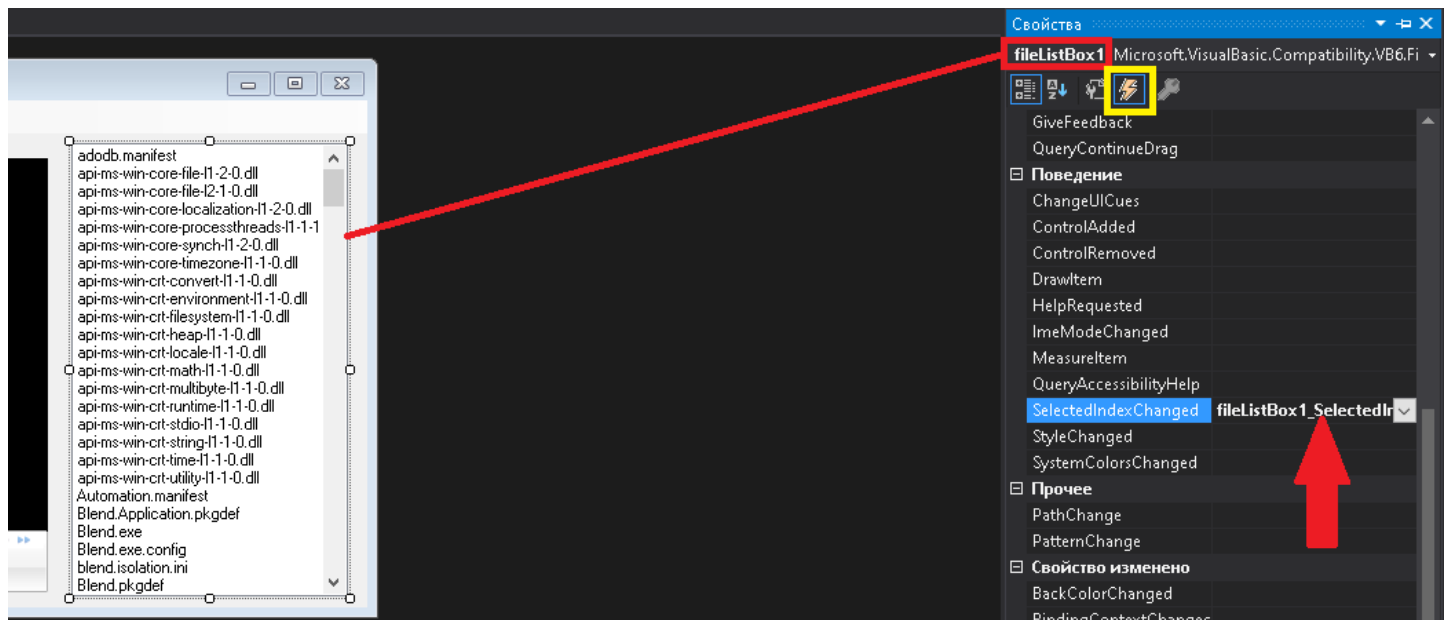
В коде файла MyForm.h кнопкой Enter сносим фигурные скобки тела метода на строки ниже него и набираем вышепоказанный код. Компилируем и тестируем.



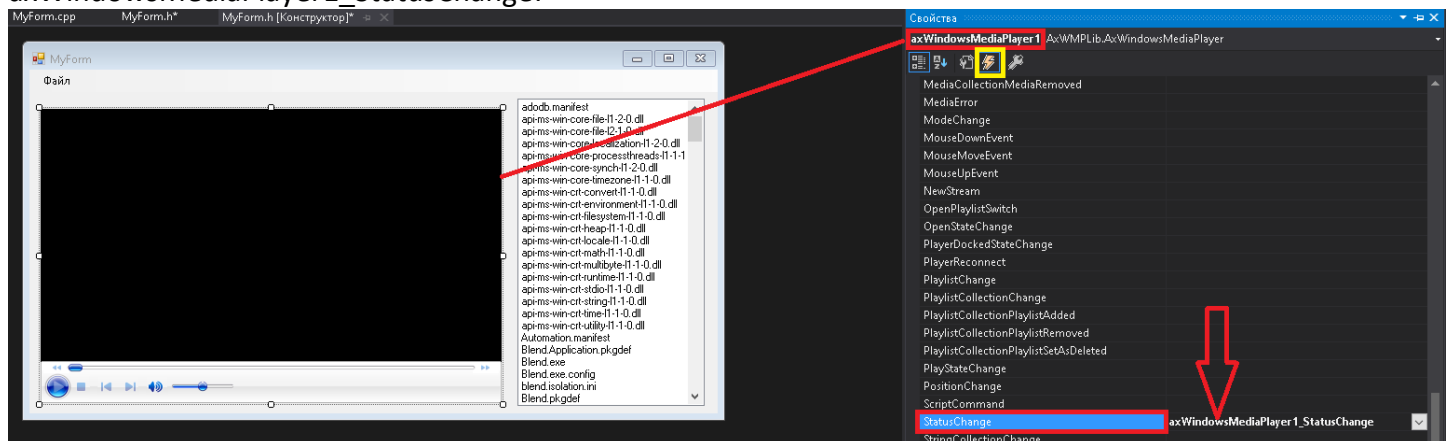
Действительно, в списке отображаются только музыкальные mp3-файлы из выбранной ранее папки:



Теперь надо выделить объект `fileListBox1` и вызвать нажатием правой кнопки по нему его окно Свойств, в котором выбрать вкладку События (Events, выглядит как «молния») и в подвкладке Поведение выбрать событие `SelectedIndexChanged`, для чего дважды кликнуть по нему справа мышью и в случае успеха откроется файл `MyForm.h` со сгенерированным обработчиком события `fileListBox1_SelectedIndexChanged` (снова откорректируйте расположение фигурных скобок его пока пустого тела). Так можно сделать любой доступный обработчик события для визуального компонента.



И аналогично сделайте обработчик события изменения статуса медиаплеера, то есть реагирования на событие запуска в плеере музыкального файла. Для `axWindowsMediaPlayer1` вкладка События, подвкладка Прочее, событие `StatusChange`, чтобы по клику мыши справа появился обработчик события `axWindowsMediaPlayer1_StatusChange`:



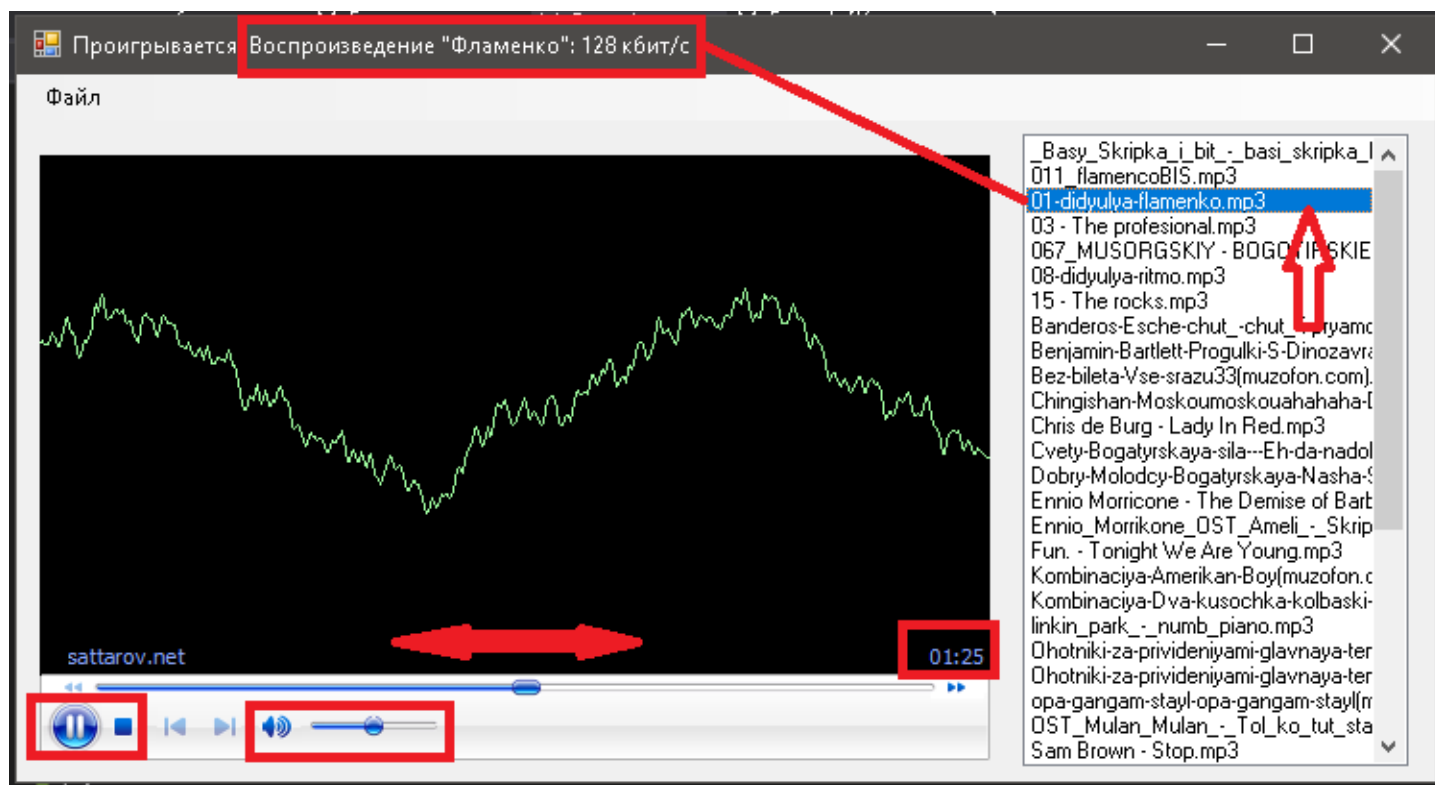
Заполняем кодом тела обработчиков событий:

```

130 #pragma endregion
131 private: System::Void ОткрытьФайлToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) //обработчик клика мыши по пункту меню "Открыть файл"
132 {
133     fileListBox1->Pattern = "*.mp3"; //оставим в списке только музыкальные файлы с расширением mp3, например
134     folderBrowserDialog1->ShowDialog(); //открыть окно просмотра папок и файлов как обязательное (модальное)
135     fileListBox1->FileName = folderBrowserDialog1->SelectedPath; //в список файлов отобразить только файлы из папки, путь к которой выбрали в окне диалога
136 } //выбора папок, причем отобразит только mp3-файлы, а если их в выбранной папке не окажется - список будет пустой
137 private: System::Void fileListBox1_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) //обработчик выбора строки из списка файлов
138 { //присвоить свойству-сеттеру задания пути к файлу для медиаплеера строку: выбранный путь, наклонные черты (на самом деле нам надо одна черта, но первая из
139     axWindowsMediaPlayer1->URL = folderBrowserDialog1->SelectedPath + "\\\" + fileListBox1->SelectedItem->ToString(); //них экранирует второй обратный слеш,
140     //чтобы он воспринимался тут как обычный char символ, а не управляющий символ, и название выбранного из списка файла, который является объектом и должен
141     //быть приведен к строковому типу данных. Получается полная строка пути к файлу, начиная от диска на ПК. Перегруженный оператор+ склеивает строки в одну
142     private: System::Void axWindowsMediaPlayer1_StatusChange(System::Object^ sender, System::EventArgs^ e) //обработчик изменения статуса медиаплеера(запуск файла
143     { //воспроизведение файла и т.д. Свойству-сеттеру Text текущего окна присвоим строку "Проигрывается: ", склеенную со строкой, возвращенной свойством-getterом
144         this->Text = "Проигрывается: " + axWindowsMediaPlayer1->status; //status. И дописывать тут ->ToString() (привести к строковому типу данных) не обязательно,
145         // поскольку свойство-getter status само возвращает строку типа String^
146     }
147 }

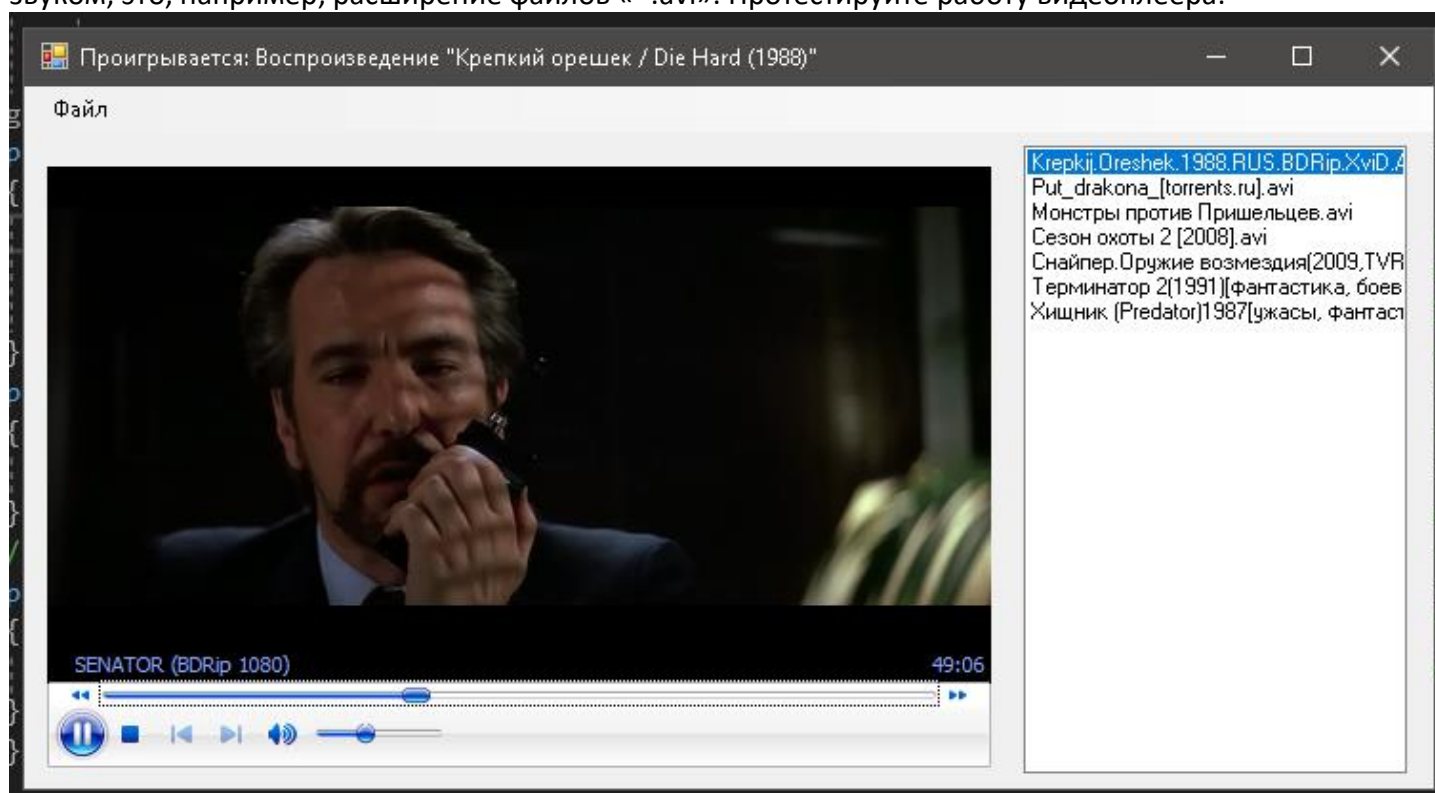
```

Тестируем приложение-плеер (выделенные кнопки работают, участки динамически перерисовываются, динамики воспроизводят музыку):



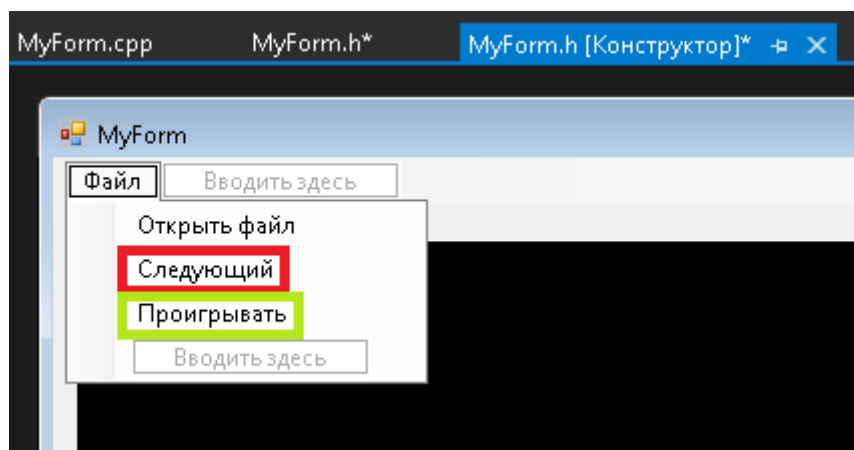
Задание 2

Перепишите в нужном месте код плеера, чтобы он воспроизводил медиафайлы (видеоряд со звуком; это, например, расширение файлов «*.avi». Протестируйте работу видеоплеера:



Задание 3

В режиме Дизайнера добавьте еще 2 подпункта контекстного меню в главное меню и дайте им названия «Следующий» и «Проигрывать»:



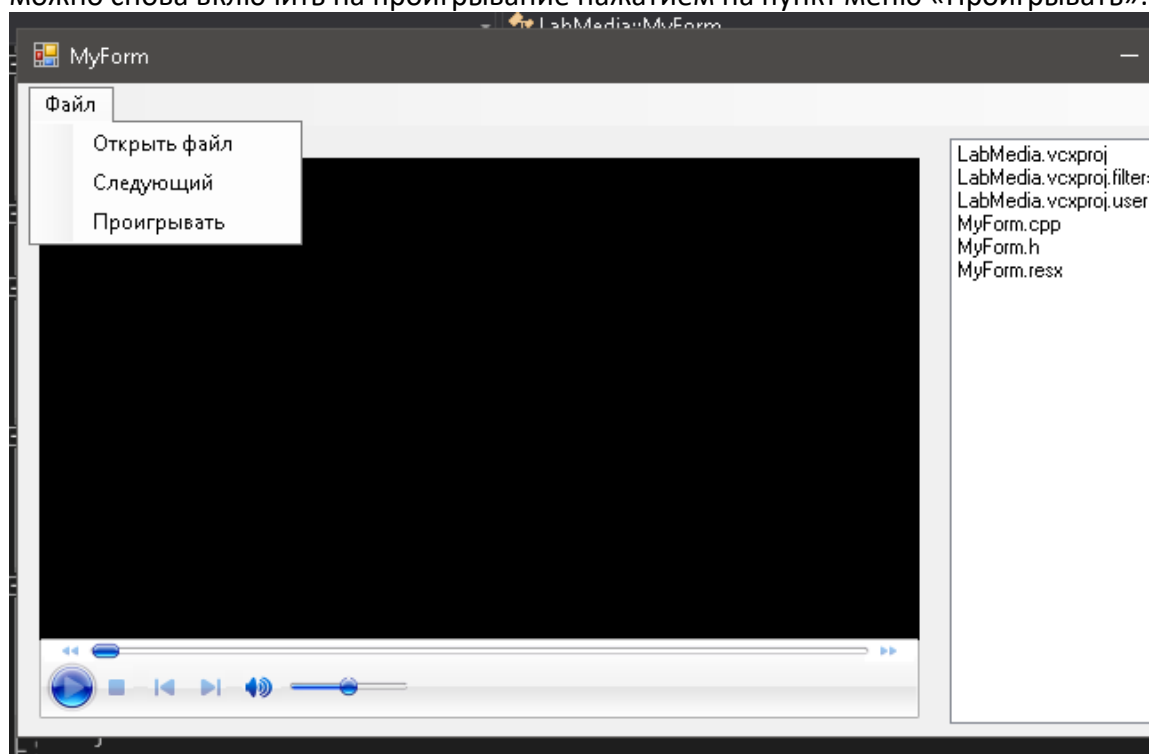
Сгенерируйте обработчики клика мыши пользователем по ним и заполните нижеследующим кодом:

```

166 // поскольку свойство-getter status само возвращает строку типа String^
167 private: System::Void следующийToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e)//обработчик клика мыши по пункту меню "Следующий"
168 {
169     try//отлов ошибок на случай выхода за диапазон списка имен файлов
170     {
171         fileListBox1->SelectedIndex += 1;//свойство SelectedIndex возвращает и задает отсчитываемый от нуля включительно индекс текущего выделенного
172         //элемента в списке System::Windows::Forms::ListBox
173         catch (System::ArgumentOutOfRangeException^ e)//если произошла ошибка выхода за диапазон индексов списка имен файлов,
174         {
175             MessageBox::Show("Список закончился.");//то вывести сообщение
176         }
177         catch (System::Exception^ e)//если любая ошибка,
178         {
179             MessageBox::Show("Ошибка: " + e->Message);//то показать сообщение с ее описанием
180         }
181     }
182 private: System::Void проигрыватьToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e)//обработчик клика мыши по пункту меню "Проигрывать"
183 {
184     try
185     {
186         //для поработы с медиаплеером напрямую, используем его свойства и методы: проиграть файл play(), остановить stop(), поставить на паузу pause() и т.д.
187         axWindowsMediaPlayer1->Ctlcontrols->play();//вызвать у медиаплеера свойство Ctlcontrols, а у него метод play() - проигрывания текущего файла
188     }
189     catch (System::Exception^ e)//если любая ошибка,
190     {
191         MessageBox::Show("Ошибка: " + e->Message);//то показать сообщение с ее описанием
192     }
193 }
194 //конец тела метода проигрыватьToolStripMenuItem_Click()
195 //конец тела класса MyForm
196 //конец тела пространства имен namespace LabMedia

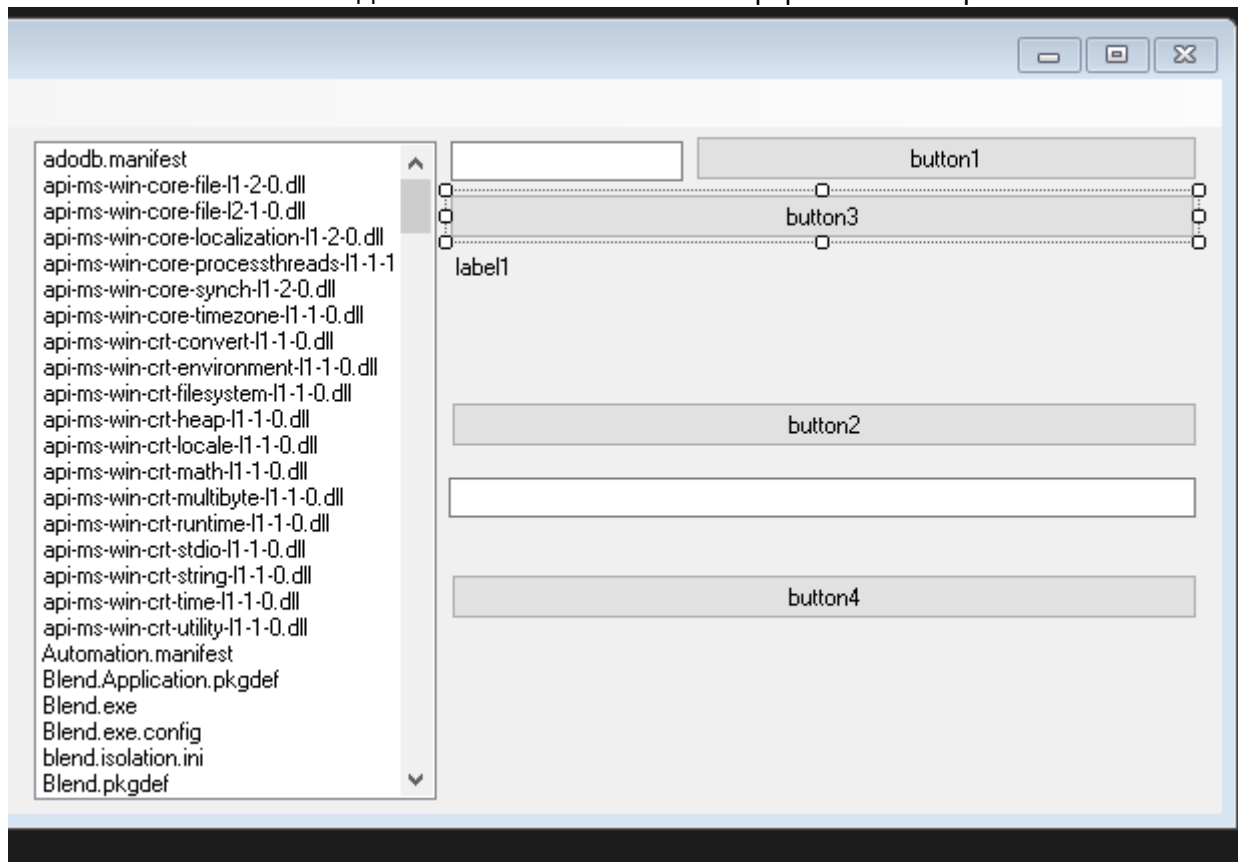
```

Скомпилируйте и протестируйте, что при вызове пункта меню «Следующий» плеер начинает проигрывать следующий по списку файл из списка fileListBox1, а если поставить на паузу файл, то его можно снова включить на проигрывание нажатием на пункт меню «Проигрывать»:



Задание 4

В ВинФормах достаточное количество вариантов списков для отображения последовательностей, массивов, например, выше была продемонстрирована работа со списком `fileListBox1`. Попробуем для работы с динамическим неуправляемым Сборщиком мусора массивом использовать 4 кнопки, 2 текстБокса и 1 лейбл. Создайте их на макете оконной формы в таком расположении:



Пусть у нас будет поле для хранения переменной – размера массива, поле-указатель на начало массива в оперативной памяти, поле-текущий индекс последнего заполненного элемента массива. Массив будет динамический неуправляемый Сборщиком мусора – управлять выделением памяти под него и удалением памяти при его очистке будем мы сами. Сделайте пустые обработчики событий нажатия на каждую из четырех кнопок `Button` и заполните их таким кодом. Защищенные поля создавайте в конце класса `MyForm`, после окончания `#pragma endregion`.

```
177 private://защищенные члены класса MyForm
178 int n = 0;//размер динамического неуправляемого Сборщиком мусора массива, по умолчанию, равен нулю
179 int* m = nullptr;//поле-казатель на целочисленный неуправляемый массив, по умолчанию, зануляем
180 int i = 0;//поле-индекс очередного элемента массива
181 private: System::Void button1_Click(System::Object^ sender, System::EventArgs e)//сгенерировать случайное число и поместить его в
182 {
183     Random^ r = gcnew Random();//в CLR-проектах вызвать random можно так, со Сборщиком мусора
184     int a = 10 + (r->Next()) % (99 - 10 + 1);//диапазон рандомизации от 10 включительно до 99 включительно
185     textBox1->Text = a.ToString();//число приводим к строковому виду и отображаем в текстБоксе1
186 }
187 private: System::Void button2_Click(System::Object^ sender, System::EventArgs e)//создать новый массив из n целых элементов
188 {
189     try
190     {
191         if (m != nullptr)//если массив уже создан
192         {
193             delete[] m;//удаляем старый
194             m = nullptr;//зануляем указатель
195             n = 0;//теперь размер массива равен нулю
196             label1->Text = "";//очищаем лейбл1 так
197             label1->Text = "Удален имевшийся массив.";//в очищенный лейбл1 выводим сообщение
198         }
199         this->n = Convert::ToInt32(this->textBox1->Text);//из текстБокса1 берем текст, приводим его к целочисленному типу и присваиваем в поле-размер динамического
200         //неуправляемого Сборщиком мусора массива. Если текст не приводится к целому числу, у нас есть обработчик ошибок try-catch
201         m = new int[n];//выделяем память под массив
202         if (!m)//if (m == nullptr)//указатель по прежнему нулевой - ни на какой массив не указывает
203         {
204             MessageBox::Show("Не удалось создать массив на " + n.ToString() + " элементов.");
205             return;//прекращаем работу метода
206         }
207         i = 0;//массив создан, но не заполнен пока, поэтому индекс текущего элемента равен нулю
208         label1->Text = "Создан массив на " + n.ToString() + " элементов.";//сообщаем пользователю
209     }
210     catch (System::FormatException^ e)//обработчик ошибки формата данных
211     {
212         MessageBox::Show(e->Message);
213     }
```

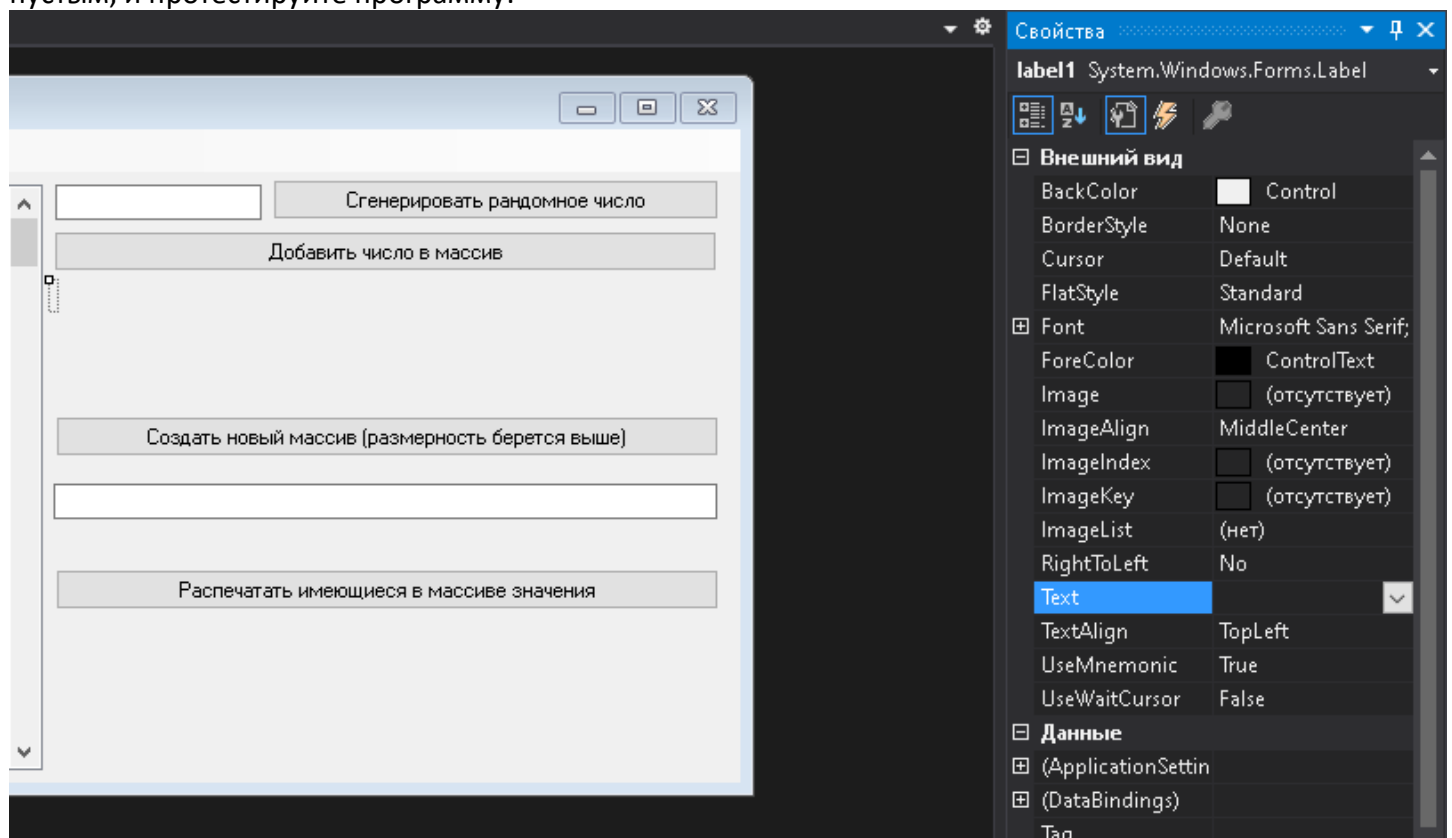


```

314 catch (System::Exception^ e)//обработчик любой ошибки
315 {
316     MessageBox::Show(e->Message);
317 }
318 }
319 private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e)//внести в массив одно новое очередное число
320 {
321     try
322     {
323         if (i < n && n > 0 && m != nullptr)//вносить новое значение в массив есть смысл только если индекс этого нового значения
324             { // меньше размера массива И размер массива больше нуля И указатель на массив не нулевой (т.е. массива нет)
325                 m[i] = Convert::ToInt32(this->textBox1->Text); //текст из текстовка1 конвертируем в целочисленный тип и помещаем в массив
326                 i++; //индекс увеличиваем для следующего элемента
327                 label1->Text = ""; //лейбл1 очищаем
328                 label1->Text = "Заполнен элемент № " + i.ToString() + " из " + n.ToString(); //сообщаем текущую ситуацию
329             }
330         else
331         {
332             MessageBox::Show("Массив размерности " + n + " заполнен или не создан.");
333         }
334     }
335     catch (System::FormatException^ e)
336     {
337         MessageBox::Show(e->Message);
338     }
339     catch (System::Exception^ e)
340     {
341         MessageBox::Show(e->Message);
342     }
343 }
344 private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e)//вывести массив в текстовке2
345 {
346     this->textBox2->ReadOnly = true; //сделать текстовк2 недоступным для редактирования пользователем (но программа может)
347     this->textBox2->Clear(); //очищаем текстовк2
348     this->textBox1->Clear(); //очищаем текстовк1
349     for (int j = 0; j < i && j < n; j++) //в цикле проходим по массиву, при этом не выходим за размерность массива и не заходим
350     { //после последнего заполненного элемента, ведь кнопку4 можно нажать заранее, пока массив еще полностью не заполнен, но
351         this->textBox2->Text += m[j].ToString() + " "; //программа должна корректно отобразить имеющиеся значения в массиве
352     } //в текстовке2 склеиваем значения из массива, приведенные к строковому типу данных, а между ними ставим пробелы
353 }
354 }; //конец тела класса MyForm
355 } //конец тела пространства имен namespace LabMedia

```

Переименуйте надписи на кнопках на более понятные пользователю, текст лейбела1 сделайте пустым, и протестируйте программу:



Тестирование:

Сгенерировать случайное число

Добавить число в массив

Создан массив на 10 элементов.

Создать новый массив (размерность берется выше)

Сгенерировать случайное число

Добавить число в массив

Создан массив на 10 элементов.

Создать новый массив (размерность берется выше)

Сгенерировать случайное число

Добавить число в массив

Заполнен элемент № 1 из 10

Создать новый массив (размерность берется выше)

Сгенерировать случайное число

Добавить число в массив

Заполнен элемент № 1 из 10

Создать новый массив (размерность берется выше)

Распечатать имеющиеся в массиве значения

Сгенерировать случайное число

Добавить число в массив

Заполнен элемент № 4 из 10

Создать новый массив (размерность берется выше)

62 97 95 89

Распечатать имеющиеся в массиве значения

77

Сгенерировать случайное число

Добавить число в массив

Заполнен элемент № 10 из 10

Создать новый массив (размерность берется выше)

62 97 95 89

Распечатать имеющиеся в массиве значения

Массив размерности 10 заполнен или не создан.

ОК

Сгенерировать случайное число

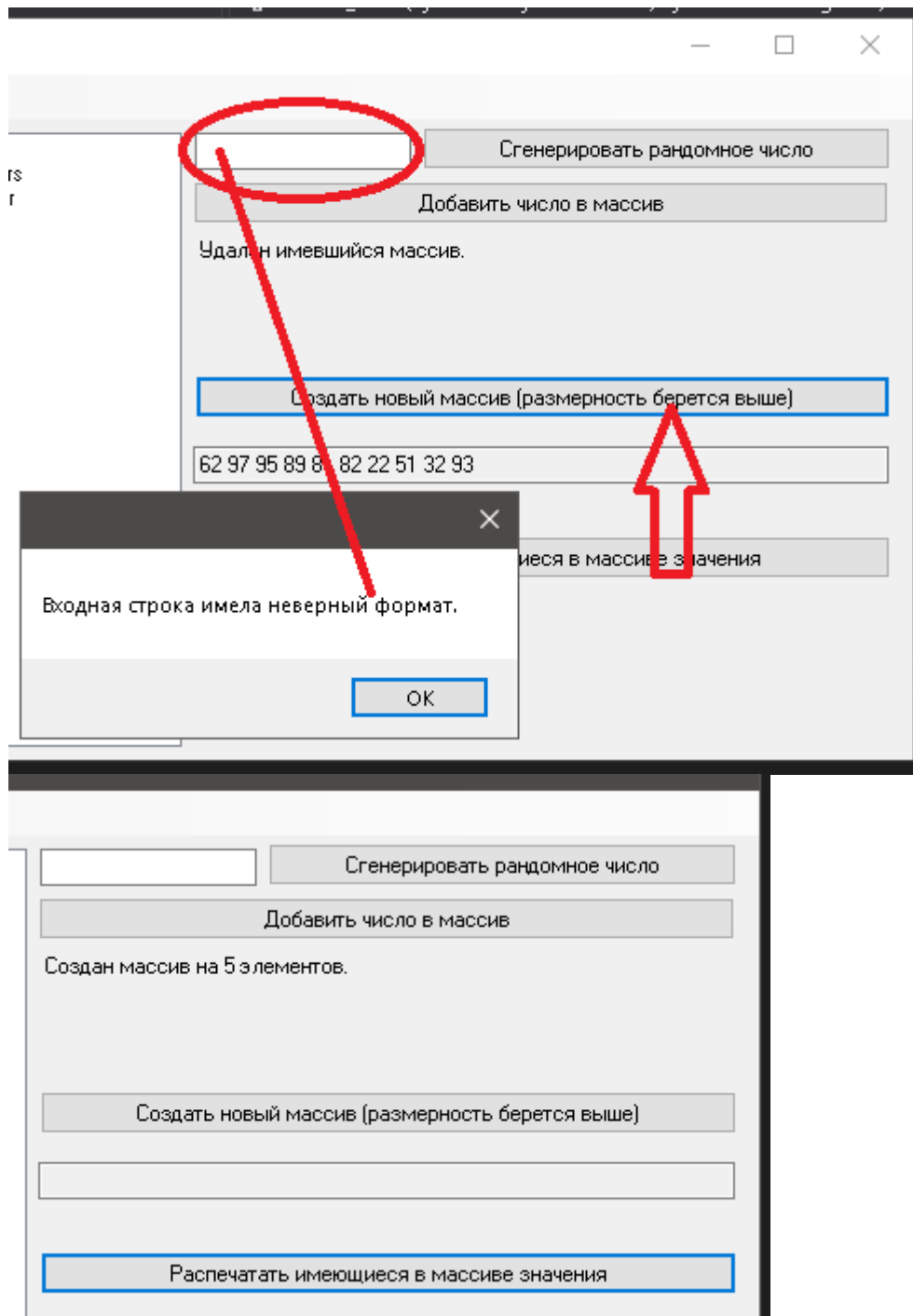
Добавить число в массив

Заполнен элемент № 10 из 10

Создать новый массив (размерность берется выше)

62 97 95 89 81 82 22 51 32 93

Распечатать имеющиеся в массиве значения



2.Порядок выполнения работы

1. Изучить теоретические сведения к лабораторной работе.
2. Разработать на языке C++ программу вывода на экран решения задачи в соответствии с вариантом индивидуального задания, указанным преподавателем.
3. Отлаженную, работающую программу сдать преподавателю. Работу программы показать с помощью самостоятельно разработанных тестов.
4. Ответить на контрольные вопросы.

Задание 5

Выполнить задания по вариантам.

Создать новый оконный проект с главным меню (MenuStrip), с помощью которого можно будет создать динамический массив, указав в окне его максимальную размерность, можно вводить в массив значения элементов от руки или генерировать их случайно в диапазоне, задаваемом пользователем, а также распечатывать и удалять массив из памяти. Пусть главное меню состоит из горизонтальных пунк-

тов меню, имеющих подпункты (если последние нужны). **Предусмотреть пункт меню для вызова отображения фамилии и имени разработчика программы.**

№	Тип данных в массиве	Что посчитать и отобразить для массива
1	Целочисленный	Среднее арифметическое (вещественное точное)
2	Короткий целочисленный	Корень из суммы (вещественное точное)
3	Длинный целочисленный	Среднее геометрическое (вещественное точное)
4	Вещественный	Количество положительных
5	Символьный	Количество букв английского алфавита
6	Логический	Количество значений ИСТИНА
7	Целочисленный	Поменять каждое значение на противоположное
8	Короткий целочисленный	Количество четных положительных
9	Длинный целочисленный	Количество НЕчетных отрицательных
10	Вещественный	Количество отрицательных
11	Символьный	Количество цифр
12	Логический	Количество значений ЛОЖЬ
13	Целочисленный	Количество четных отрицательных
14	Короткий целочисленный	Количество НЕчетных положительных
15	Длинный целочисленный	Количество значений, больших значения, указанного пользователем
16	Вещественный	Количество значений в диапазоне от -3 до +5
17	Символьный	Количество символов – знаков препинания
18	Логический	Распечатать, каких значений больше
19	Целочисленный	Сумму значений поделить на их произведение (вещественное точное)
20	Короткий целочисленный	Количество значений, меньших значения, указанного пользователем
21	Длинный целочисленный	Количество значений, НЕравных нулю
22	Вещественный	Количество значений вне диапазона от -9.5 включительно до +0.99 не включительно
23	Символьный	Указать, чего больше: английских букв или цифр
24	Логический	Распечатать, каких значений меньше
25	Целочисленный	Произведение всех значений массива поделить на их сумму (вещественное точное)
26	Короткий целочисленный	Количество значений, кратных 7 (семи)
27	Длинный целочисленный	Количество значений, НЕкратных 5 (пяти)
28	Вещественный	Количество равных нулю
29	Символьный	Общее количество символов: английских букв и цифр
30	Логический	Поменять каждое значение на противоположное

3. Контрольные вопросы

1. Зачем нужно и как создать Главное меню?
2. С помощью каких графических элементов можно ввести значения в массив?
3. С помощью каких графических элементов можно вывести значения из массива на оконную форму?
4. Что такое COM-компонент? Приведите примеры и их назначение.
5. Какие типы и расширения файлов можно воспроизводить в MS Windows Media Player?
6. Что такое FolderBrowserDialog и зачем он нужен?

7. С помощью чего можно отображать списки различных данных и объектов в оконной форме?
8. Где в режиме графического Конструктора отображаются невидимые элементы, помещенные на оконную форму?
9. В чем разница управляемого и неуправляемого динамического массивов? Кем (чем) управляется каждый из них?
10. Какие диалоговые окна можно вызвать в оконном приложении?

Литература

Зиборов, В. MS Visual C ++2010 в среде NET (Библиотека программиста) / В. Зиборов.— СПб.: Питер, 2012.

Преподаватель

Шаляпин Ю.В.

Рассмотрено на заседании цикловой
комиссии ПОИТ № 10

Протокол № ____ от « ____ » _____ 2021 г.

Председатель ЦК _____