Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

ОТЧЕТ

По лабораторной работе №1

«**ОСНОВЫ ПРОГРАММИРОВАНИЯ В С#.NET .РАЗРАБОТКА ПРИЛОЖЕНИЙ НА ОСНОВЕ ФОРМЫ**»

|  |  |
| --- | --- |
| Проверил:  Выполнил: | А. Л. Гончаревич  ст. гр. 120603  С. А. Карапетян |

Минск 2024

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc159025931)

[1 Постановка задачи 4](#_Toc159025932)

[2 Теоретическая часть 5](#_Toc159025933)

[3 Практическая часть 9](#_Toc159025934)

[Заключение 13](#_Toc159025935)

# ВВЕДЕНИЕ

Целью лабораторной работы является разработка оконного приложения на основе *Windows* *Forms* с использованием языка программирования *C#* и платформы *.NET Framework*, которая предназначена для работы на операционной системе *Windows*. Основными компонентами платформы *.NET Framework* являются *Common Language Runtime* (*CLR*), который обеспечивает управление памятью, выполнение кода, обработку исключений и управление потоками, и библиотека классов, предоставляющая набор готовых компонентов и функций для разработки приложений. Также *.NET Framework* поддерживает несколько языков программирования, включая *C#*, благодаря чему разработчики могут выбрать язык, наиболее удобный для них, и использовать его для создания приложений на платформе .*NET*.

*Windows Forms* (*WinForms*) представляет собой библиотеку классов в *.NET Framework*, которая обеспечивает разработчикам возможность создавать графические пользовательские интерфейсы для приложений, работающих в операционной системе *Windows*. Основные компоненты *Windows Forms* включают в себя формы (*Forms*), элементы управления (*Controls*), макеты (*Layouts*), а также события и обработку событий. *Windows Forms* облегчает процесс создания и развертывания графических приложений для платформы *Windows*. Она предоставляет разработчикам удобный интерфейс программирования и инструменты для создания пользовательских интерфейсов, обработки событий и взаимодействия с другими компонентами *.NET Framework*.

# 1 Постановка задачи

Изучить базовые возможности языка программирования *C#*. Познакомиться с принципами и методами разработки оконных приложений на основе формы. Изучить возможности библиотеки классов *Windows Forms* платформы *.NET Framework*.

Научиться создавать графические пользовательские интерфейсы для приложений с помощью *Windows Forms*. Познакомиться с элементами формы, научиться добавлять, изменять и удалять элементы, а также создавать события и обработчики событий. Изучить и научиться работать с основными компонентами формы, а также их свойствами.

Продемонстрировать возможности языка программирования *C#* и библиотеки классов *Windows Forms* платформы *.NET Framework*. Использовать полученные знания для реализации оконного приложения.

Создать оконное приложение для просмотра файлов мультимедиа, в котором будут реализованы следующие возможности: выбор файлов для воспроизведения из репозитория компьютера, создание плейлистов, регулирование громкости воспроизведения файла, а также отслеживание и регулирование процесса проигрывания файлов мультимедиа.

# 2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

*Windows Forms* (*WinForms*) представляет собой библиотеку классов в .*NET Framework*, которая обеспечивает разработчикам возможность создавать графические пользовательские интерфейсы для приложений, работающих в операционной системе *Windows*.

Панель элементов (*Toolbox*) в *Windows* *Forms* представляет собой панель, содержащую набор предопределенных элементов управления и компонентов, которые разработчик может использовать при создании пользовательского интерфейса. В панели элементов доступны различные категории элементов управления, такие как кнопки, метки, списки, полосы прокрутки и другие.

Разработчик может перетаскивать элементы управления из панели элементов на форму, чтобы добавить их в пользовательский интерфейс. Когда элемент управления добавлен на форму, разработчик может настроить его свойства, такие как размер, положение, текст, стиль и другие атрибуты. Это возможно реализовать при помощи панели свойств формы (*Properties*) в *Visual Studio*, которая отображает доступные свойства выбранного элемента управления.

Использование вкладок панели элементов и панели свойств помогает разработчику управлять и настраивать элементы управления в *Windows Forms*, обеспечивая удобство и эффективность в процессе разработки пользовательского интерфейса.

*Windows Forms* облегчает процесс создания и развертывания графических приложений для платформы Windows. Она предоставляет разработчикам удобный интерфейс программирования и инструменты для создания пользовательских интерфейсов, обработки событий и взаимодействия с другими компонентами *.NET Framework*.

Примерами предопределенных элементов управления и компонентов, которые используются при создании оконного приложения для просмотров файлов мультимедиа, являются:

1. «*TrackBar*»: это элемент управления, который позволяет пользователю выбирать значение из некоторого диапазона, перемещая ползунок по горизонтальной или вертикальной шкале.
2. «*Label*»: представляет собой текстовую метку или надпись, которая используется для отображения статического текста на форме. Он не является интерактивным и предназначен для отображения информации или инструкций для пользователя.
3. «*Button*»: является интерактивным элементом управления, который представляет собой кнопку с текстом или изображением. Он используется для инициирования действий или обработки событий при щелчке на него пользователем.
4. «*Timer*»: компонент, который позволяет разработчику создавать события, которые будут вызываться через определенные промежутки времени.
5. «*OpenFileDialog*»: представляет собой диалоговое окно, которое позволяет пользователю выбрать файл для открытия. Он предоставляет удобный способ для получения пути к выбранному файлу в пользовательском интерфейсе.
6. «*MediaPlayer*»: представляет собой компонент, который обеспечивает возможность воспроизведения мультимедийных файлов, таких как аудио или видео.
7. «*Panel*»: представляет собой контейнер, который позволяет группировать и управлять другими элементами управления внутри себя. Панель обычно используется для создания областей на форме, в которых можно размещать и организовывать другие элементы.

Панель свойств предоставляет удобный способ для разработчика настраивать свойства элементов управления без необходимости написания кода вручную. При выборе элемента управления на форме, в панели свойств отображаются все доступные свойства и их значения, которые могут быть изменены.

Основные свойства элемента управления «*TrackBar*»:

1. *Minimum*: определяет минимальное значение ползунка. Ползунок не может быть установлен ниже этого значения.
2. *Maximum*: определяет максимальное значение ползунка. Ползунок не может быть установлен выше этого значения.
3. *Value*: определяет текущее значение ползунка. Может быть установлено в диапазоне от минимального до максимального значения.
4. *SmallChange*: это свойство определяет значение, на которое изменяется ползунок при использовании стрелок клавиатуры или щелчка на стрелках на ползунке.
5. *TickFrequency*: определяет частоту отображения меток на полосе ползунка. Метки помогают пользователю ориентироваться в диапазоне значений.
6. *Orientation*: определяет ориентацию ползунка. Может быть *Horizontal* (горизонтальный) или *Vertical* (вертикальный).
7. *TickStyle*: определяет стиль отображения меток на полосе ползунка. Варианты включают *None* (без меток), *TopLeft* (метки сверху или слева) и *BottomRight* (метки снизу или справа).
8. *AutoSize*: определяет, будет ли размер ползунка автоматически изменяться в соответствии с его содержимым.

Эти свойства позволяют разработчикам настраивать поведение и внешний вид элемента управления «*TrackBar*» и управлять выбором значения пользователем в заданном диапазоне.

Для создания события необходимо определить его в классе с помощью ключевого слова «*event»*, например, «*public event EventHandler MyEvent*», а затем написать обработчик этого события, который будет вызываться при его возникновении.

В *C#* существует понятие частичного класса. Частичный класс ‒ это класс, разделенный на несколько фрагментов (частей), каждая из которых может находиться в разных файлах. Все эти фрагменты объединяются компилятором в один класс.

Частичные классы обычно используются для упрощения работы с крупными классами или для организации кода в больших проектах, когда несколько разработчиков работают над одним классом.

Классы-контейнеры являются одним из ключевых элементов программирования, используемых для управления группами объектов или данных. Такие классы обеспечивают структурирование, хранение и управление наборами объектов определенного типа в рамках приложения. Класс-контейнер предоставляет методы для управления этой коллекцией, такие как добавление, удаление, поиск и изменение элементов. Примеры классов-контейнеров включают списки, массивы, стеки, очереди и т. д.

Для динамического создания компонента необходимо создать его экземпляр, используя оператор «*new»*, а затем установить его свойства. После этого добавить созданный компонент на форму с помощью метода добавления, предоставленного используемым фреймворком. Например, «*Controls*.*Add»* для *Windows* *Forms*.

Размещение элементов на форме обычно выполняется с помощью менеджера компоновки (*layout manager*). Менеджер компоновки отвечает за расположение и размер элементов на форме в зависимости от заданных правил компоновки, определяет расположение и размер элементов на основе их свойств и правил компоновки, таких как док-положение, якорь, размеры и т. д.

Невизуальный класс-контейнер ‒ это класс, который не представляет собой элемент пользовательского интерфейса, а скорее служит для управления и организации данных или других объектов. Он обычно используется для хранения и управления коллекциями объектов определенного типа или для выполнения операций над этими объектами без непосредственного взаимодействия с пользователем.

Основными характеристиками невизуальных классов-контейнеров являются управление коллекциями данных, организация и структурирование данных, безопасность данных и согласованность, выполнение операций над данными.

Примеры невизуальных классов-контейнеров включают в себя классы для управления списками пользователей, классы для доступа к базам данных и выполнения запросов, классы для управления файлами и директориями в файловой системе, классы для управления сетевыми соединениями и многое другое. Эти классы обычно не имеют непосредственного отображения на экране пользователя, но играют важную роль в управлении данными и выполнении функций приложения.

# 3 ПРАКТИЧЕСКАЯ ЧАСТЬ

После запуска приложения открывается стартовое окно, это показано на рисунке 3.1. В этом окне расположены медиа-плеер, в котором будут воспроизводиться выбранные пользователем медиафайлы, необходимые для работы с приложением кнопки открытия файлов, воспроизведение и пауза проигрывания, кнопка удаления медиафайла из плейлиста, кнопка добавления нового медиафайла в уже существующий плейлист, кнопки выбора следующего или предыдущего медиафайла в плейлисте, ползунок регулирования громкости воспроизведения файлов мультимедиа и кнопка для переключения между режимами проигрывания, такими как проигрывание файлов в случайном порядке, циклическое проигрывание одного файла и циклическое проигрывание плейлиста.

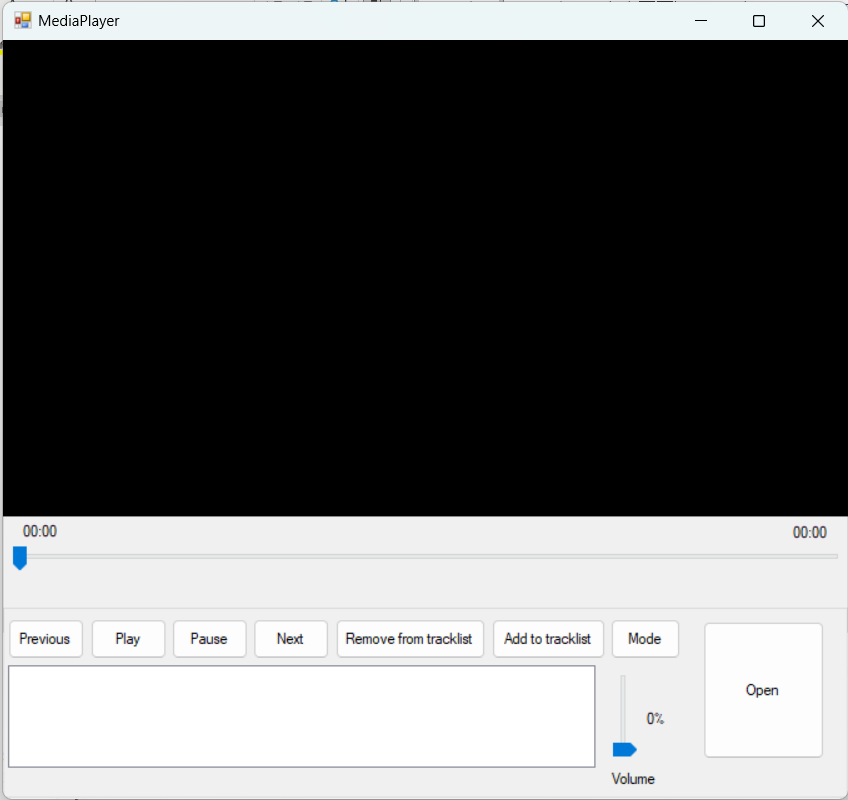


Рисунок 3.1 – Стартовое окно

Для выбора файлов проигрывания, необходимо нажать кнопку «*Open*». Открывается проводник, в котором можно выбрать необходимые медиафайлы. После выбора необходимо нажать кнопку «Открыть». Выбранные медиафайлы будут загружены в текущий плейлист. Окно выбора медиафайлов представлено на рисунке 3.2.

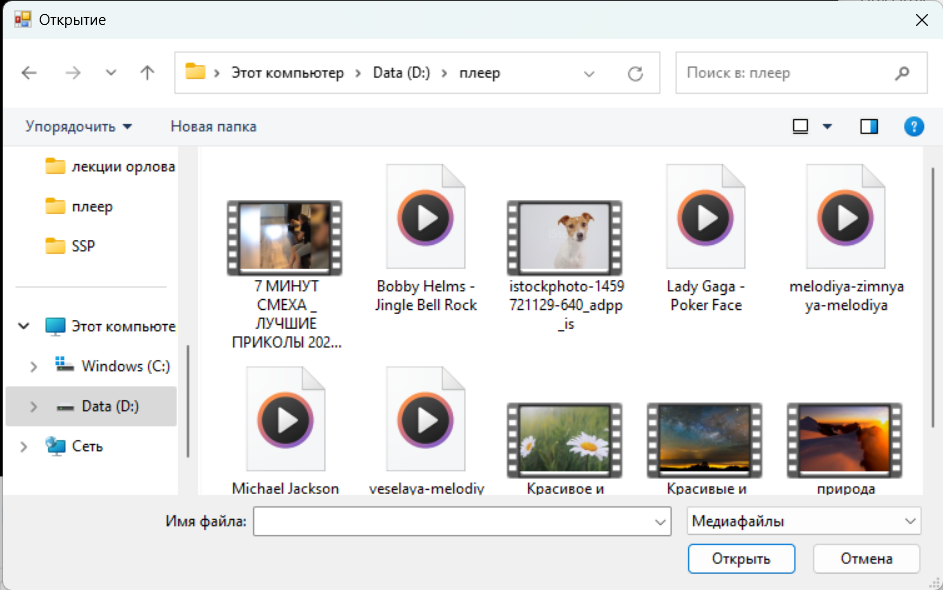


Рисунок 3.2 – Окно выбора медиафайлов

Воспроизведение медиафайлов начинается автоматически после выбора файлов в плейлист. Окно приложения с выбранными медиафайлами представлено на рисунке 3.3. Остановить воспроизведение можно, нажав кнопку «*Pause*». Для возобновления воспроизведения медиафайла, необходимо нажать кнопку «*Play*». При нажатии на кнопку «*Next*» воспроизводится следующий в списке файл мультимедиа. Для перехода к предыдущему медиафайлу необходимо нажать кнопку «*Previous*». Регулировать громкость воспроизведения медиафайла можно, двигая вертикальный ползунок в нижней части экрана. По умолчанию при воспроизведении медиафайла значение громкости устанавливается равной 15% от общей громкости, установленной на устройстве. Для перемотки воспроизводящегося медиафайла необходимо воспользоваться горизонтальным ползунком. В зависимости от положения ползунка, изменяется текущий момент воспроизведения выбранного медиафайла.

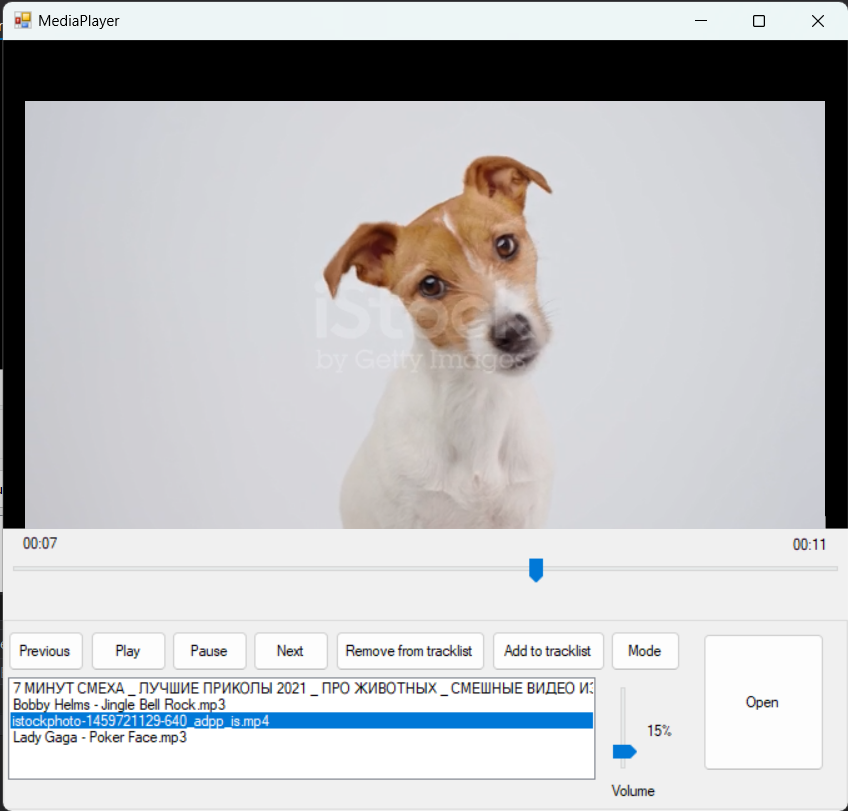


Рисунок 3.3 – Окно с выбранными медиафайлами

Кнопка «*Mode*» предназначена для переключения между режимами воспроизведения медиафайлов. Пользователю доступно три режима воспроизведения:

* «*Repeat*»: при активации этого режима воспроизведение текущего медиафайла будет повторяться;
* «*Shuffle*»: при активации этого режима воспроизведение файлов плейлиста будет проводиться в хаотичном порядке;
* «*Loop*»: при активации этого режима зацикливается воспроизведение плейлиста.

Для добавления медиафайла в текущий плейлист необходимо нажать кнопку «*Add to tracklist»*. Откроется окно выбора файлов для добавления к текущим медиафайлам плейлиста. После выбора необходимых файлов необходимо нажать кнопку «Открыть». Это продемонстрированно на рисунке 3.4.

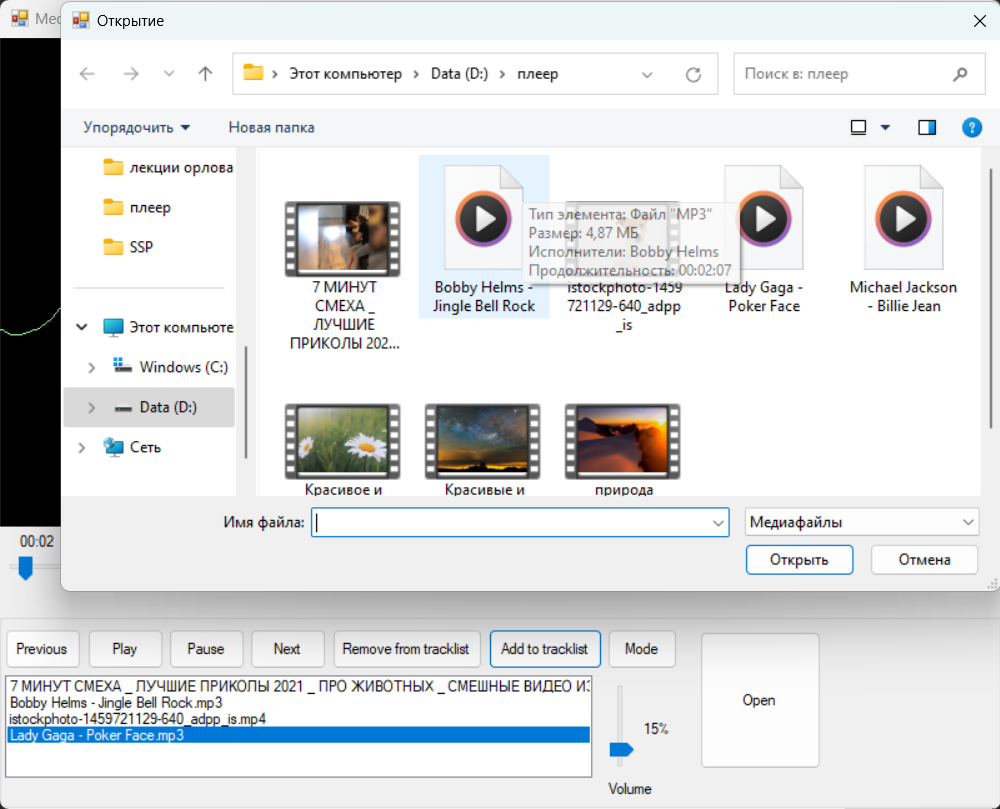


Рисунок 3.4 – Добавление медиафайлов в плейлист

Для удаления медиафайла из текущего плейлиста необходимо выбрать файл, который необходимо удалить, и нажать кнопку «*Remove from tracklist»*.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы изучены базовые возможности языка программирования *C#*, а также принципы и методы разработки оконных приложений на основе формы. Платформа *.NET Framework* предоставляет разработчикам удобные инструменты и библиотеки классов для создания графических пользовательских интерфейсов на операционной системе *Windows*.

Рассмотрены компонент платформы .*NET Framework* *Common Language Runtime* (*CLR*) и библиотека классов. *CLR* обеспечивает управление памятью, выполнение кода, обработку исключений и управление потоками, что значительно упрощает разработку и обеспечивает безопасность приложений. Библиотека классов предоставляет разнообразные компоненты и функции для разработки приложений, что позволяет разработчикам сосредоточиться на реализации основной функциональности приложения.

Изучена библиотека классов *Windows Forms* (*WinForms*). Как часть .*NET Framework*, она предоставляет удобный интерфейс программирования и набор компонентов для создания графических пользовательских интерфейсов. Было установлено, что *Windows* *Forms* облегчает процесс создания и развертывания графических приложений на платформе *Windows*. Основные компоненты *Windows Forms*, такие как формы, элементы управления, макеты и обработка событий, предоставляют разработчикам мощные инструменты для создания интерактивных и отзывчивых пользовательских интерфейсов.

Разработано оконное приложение для просмотра файлов мультимедиа с такими функциями, как выбор медиафайлов для воспроизведения, регулирование громкости воспроизведения файла, добавление и удаление медиафайлов, отслеживание и регулирование процесса проигрывания и переключение между различными режимами воспроизведения.