Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *Сєрова Даниїла Олексійовича*
(ПІБ)

академічної групи *122-18ск-2*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: *Розробка Media Player на C# з використанням Bunifu UI framework*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	*доц. Приходченко С.Д.*			
розділів:				
спеціальний	*доц. Приходченко С.Д.*			
економічний	*проф. Вагонова О.Г.*			
Рецензент				
Нормоконтролер	*доц. Гуліна І.Г.*			

Дніпро
2021

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

_____ І.М. Удовик
(підпис) (прізвище, ініціали)

« » _____ 2021 року

ЗАВДАННЯ
на кваліфікаційну роботу ступеня

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *122-18ск-2* *Сєрова Даниїла Олексійовича*
(група) (прізвище та ініціали)

тема кваліфікаційної роботи *Розробка Media Player на C#*
з використанням Bunifu UI framework

затверджена наказом ректора НТУ «ДП» від *07.06.2021* № *317-с*

Розділ	Зміст виконання	Термін виконання
Спеціальний	***На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми***	***24.05.2021 р.***
Економічний	***Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки***	***27.05.2021 р.***

Завдання видав _____ _____
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання _____ _____
(підпис) (прізвище, ініціали)

Дата видачі завдання: *14.01.2021 р.*

Термін подання кваліфікаційної роботи до ЕК: *11.06.2021 р.*

РЕФЕРАТ

Пояснювальна записка: 84 с., 23 рис., 3 дод., 20 джерел.

Об'єкт розробки: Медіаплеєр з використанням фреймворку Bunifu .

Мета кваліфікаційної роботи: розробка медіаплеєру який надаватиме користувачам зручним і функціональний інтерфейс з унікальним дизайном.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформа для розробки, виконано проектування і розробка програми, описана робота програми, алгоритм і структура її функціонування, а також виклик та завантаження програми, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні медіаплеєру, що що дозволяє функціонально користуватися програмою та надає можливість комфортного перегляду аудіо відео файлів.

Актуальність розробленої програми полягає в створенні такого медіаплеєру, який матиме попит у всій аудиторії .

Список ключових слів: МЕДІАПЛЕЄР , КОМП’ЮТЕР, ІНФОРМАЦІЙНА СИСТЕМА, ОБЛІК, АЛГОРИТМ, ПРОЕКТУВАННЯ, МЕНЮ, ВКЛАДКА, ПРОГРАМА.

ABSTRACT

Explanatory note: 84 p., 23 img., 3 apps., 20 sources.

Object of development: Media player using the Bunifu framework.

The purpose of the qualification work: to develop a media player that will provide users with a user-friendly and functional interface with a unique design.

The introduction examines the current problem statement, specifies the purpose of the qualification work and the area of its application, justifies the relevance of the topic and specifies the problem statement.

In the first section carries out the analysis of the subject area, determines the relevance of the task and the dedication of the development, creates task statement, the software and hardware requirements of the product, specifies technologies and tools for development.

In the second section analyzes the existing solutions, chose the platform for development, creates design and finish development of the product, describes the algorithm, structure and architecture solutions in the system, defines the input and output data, describes characteristics of the technical resources used, describes how to run a program, features of user interaction, differences between serve and client parts of product.

The economic section defines the complexity of the information system, calculates the cost of work on creating the application and defines the time for its creation.

The practical value lies in creating a media player that provides an opportunity to spend time with pleasure.

The relevance of the developed program is to create a media player that will be in demand among the entire audience .

Keywords: MEDIA PLAYER, COMPUTER, INFORMATION SYSTEM, ACCOUNTING, ALGORITHM, DESIGN, MENU, TAB, PROGRAM.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

VS — Visual Studio

OS — operation system;

IT — information technology;

API — application programming interface

UI — User Interface;

EX — User Experience;

CLI — Command Line Interface

IL — Instruction List

ПЗ — програмне забезпечення;

ПК — персональний комп’ютер;

ІС — інформаційна система;

ЕОМ — електронно-обчислювальна машина;

ЗМІСТ

ВСТУП

Завдання даної кваліфікаційної роботи та об'єкт його діяльності безпосередньо пов'язані з напрямом підготовки та відповідає узагальненій тематиці кваліфікаційних робіт і переліку зазначених виробничих функцій, типових задач діяльності, умінню та компетенціям, якими повинні володіти бакалаври напряму 122 «Комп’ютерні науки».

Тематикою кваліфікаційної роботи є проектування та розробка медіаплеєру на C# з використанням Bunifu UI.

Метою кваліфікаційної роботи є вивчення засобів роботи з багатьма підсистемами операційної системи. Перш за все це події вводу в системі та взаємодія з ними. Ці навички є дуже актуальними в сфері розробки програмного забезпечення для контролю доступу та моніторингу активності процесів та користувачів. Також вони допоможуть краще зрозуміти механізми роботи об’єктами та їх взаємодію на досить низькому рівні.

Дана робота дозволяє познайомитись з методами розробок WinForm додатків та їх розуміння.

Отже задачею даної кваліфікаційної роботи є проектування та розробка медіалеєру, що забезпечує користувача якісним медіаплеєром з футуристичним і зручним інтуїтивно зрозумілим інтерфейсом.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Медіаплеєр - комп'ютерна програма, призначена для відтворення файлів мультимедіа-контенту. Один різновид медіаплеєрів призначен для відтворення тільки аудіо - або ж відеофайлів, і які називаються, відповідно, - аудіоплеєр і відеоплеєр. В основному всі відеоплеєри діляться на кілька типів: прості (для роботи на слабких ПК), універсальні і спеціалізовані (специфічні і професійні програми для вирішення унікальних завдань). Приблизно за таким же принципом діляться і аудіоплеєри. Розробники таких плеєрів прагнуть зробити їх якомога зручнішими для відтворення відповідних форматів.

Інший різновид програм-медіаплеєрів підтримують як аудіо так і відео і називається Мультимедіа-центри.

Більшість сучасних операційних систем за замовчуванням містять в своєму складі медіаплеєри: наприклад, Windows Media Player, Mac OS X-QuickTime Player (для відтворення відео у форматі QuickTime) і iTunes (для деяких інших форматів), Linux — Amarok, Rhythmbox або інші (в залежності від дистрибутива).

Але таким медіаплеєрам бракує сучасного живого футуристичного інтерфейсу.

Варто розглянути існуючі аналоги та виявити їх переваги та недоліки. Для дослідження були обрані такі відомі інтерфейси програмування додатків як WinForms та WPF.

Winforms згадується як Windows Forms. Це графічний користувальницький інтерфейс для настільних додатків .NET Framework. Має набір керованих

бібліотек в .NET framework. Він пропонує велику клієнтську бібліотеку для надання інтерфейсу для доступу до елементів графічного інтерфейсу та графічних інтерфейсів власних вікон з керованого коду.

WPF скорочено позначається як середовище представлення Windows. Спочатку він був випущений Microsoft з .NET Framework 3.0 в 2006 році. Це середовище графічного інтерфейсу користувача для створення додатків Windows. WPF - це більше, ніж просто оболонка, це частина. NET framework. Він містить суміш керованого і некерованого коду.

Одним з найважливіших відмінностей між WinForms і WPF є той факт, що в WinForms інтерфейс користувача це всього лише графічний слой, що використовує стандартні елементи управління Windows (наприклад TextBox), а WPF-інтерфейс, в свою чергу, побудований «з чистого аркуша», не спираючись на стандартні елементи.

WinForm. У додатку Windows форми Windows надають оболонку, що складається з набору класів, для розробки додатків Windows, і кожен елемент управління в додатку форми Windows є конкретним екземпляром класу. Він надає різні елементи керування, такі як текстові поля, кнопки, мітки та веб-сторінки, а також параметри для створення настроюваного елемента керування. Для цього в Visual Studio є інструмент конструктора форм Windows, який обробляє елементи управління у формі і впорядковує їх відповідно до бажаного макетом для додавання коду для обробки подій.

У формах Windows налаштування програми - ще одна функція для створення, зберігання та обслуговування інформації. Клас форми Windows можна розширити за допомогою успадкування для розробки інфраструктури програми, яка забезпечує абстракцію і можливість повторного використання коду. Форми повинні бути компактними з елементами управління на обмеженому розмірі. Форми можуть бути розбиті на шматки, упаковані в збірки, які можуть автоматично оновлюватися. Розробка програми забезпечує

масштабованість і гнучкість з легкістю для налагодження та обслуговування. Форми Windows не можуть бути передані через межі домену програми.

Переваги:

– Технологія старіша і, відповідно, краще випробувана і протестована;

– На даний момент існує величезна безліч готових елементів управління, які можна купити або використовувати безкоштовно;

– З точки зору написання, дизайнер Visual Studio краще пристосований до WinForms, так як в WPF більше необхідно робити самому.

WPF. Основними компонентами архітектури WPF є структура подання, ядро подання. У WPF елементи інтерфейсу користувача розроблені в XAML, а поведінка може бути реалізована на процедурній мові. З XAML в WPF програмісти можуть працювати паралельно з дизайнерами. WPF - це потужне середовище для створення додатків Windows, що володіє такими чудовими функціями, як прив'язка даних, медіа-сервіси, шаблони, анімація, direct3D і альтернативний введення.

Розробка додатків WPF може бути виконана за допомогою інструментів Microsoft, таких як Visual Studio і Expression Blend. VS в основному використовується розробником для створення програми WPF, а Blend в основному використовується розробниками для додатків WPF.

Переваги:

– Вона новіша і, відповідно, відповідає сучасним стандартам розробки;

– Microsoft використовує її в багатьох своїх додатках, наприклад Visual Studio;

– Це більш гнучка система, можна зробити більше, без написання або покупки готових елементів управління;

– За допомогою XAML можна легко створювати і редагувати GUI, дозволяє розділити роботу дизайнера (XAML) і програміста (C#, VB.NET та ін.).

Bunifu Ui. Платформа Bunifu допомагає розробникам програмного забезпечення і компаніям по всьому світу створювати сучасні, вражаючи програмні інтерфейси і користувацький досвід. Продукт підтримує WinForms для C# і VB.NET розробники. Функції ретельно продумані розробником, щоб забезпечити швидкість і простоту.

Framework - програмна платформа, що визначає структуру програмної системи програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту.

Фреймворк відрізняється від поняття бібліотеки тим, що бібліотека може бути використана в програмному продукті просто як набір підпрограм близької функціональності, не впливаючи на архітектуру програмного продукту і не накладаючи на неї ніяких обмежень. У той час як фреймворк диктує правила побудови архітектури додатка, задаючи на початковому етапі розробки поведінку за замовчуванням - «каркас», який потрібно буде розширювати і змінювати, відповідно до зазначених вимог.

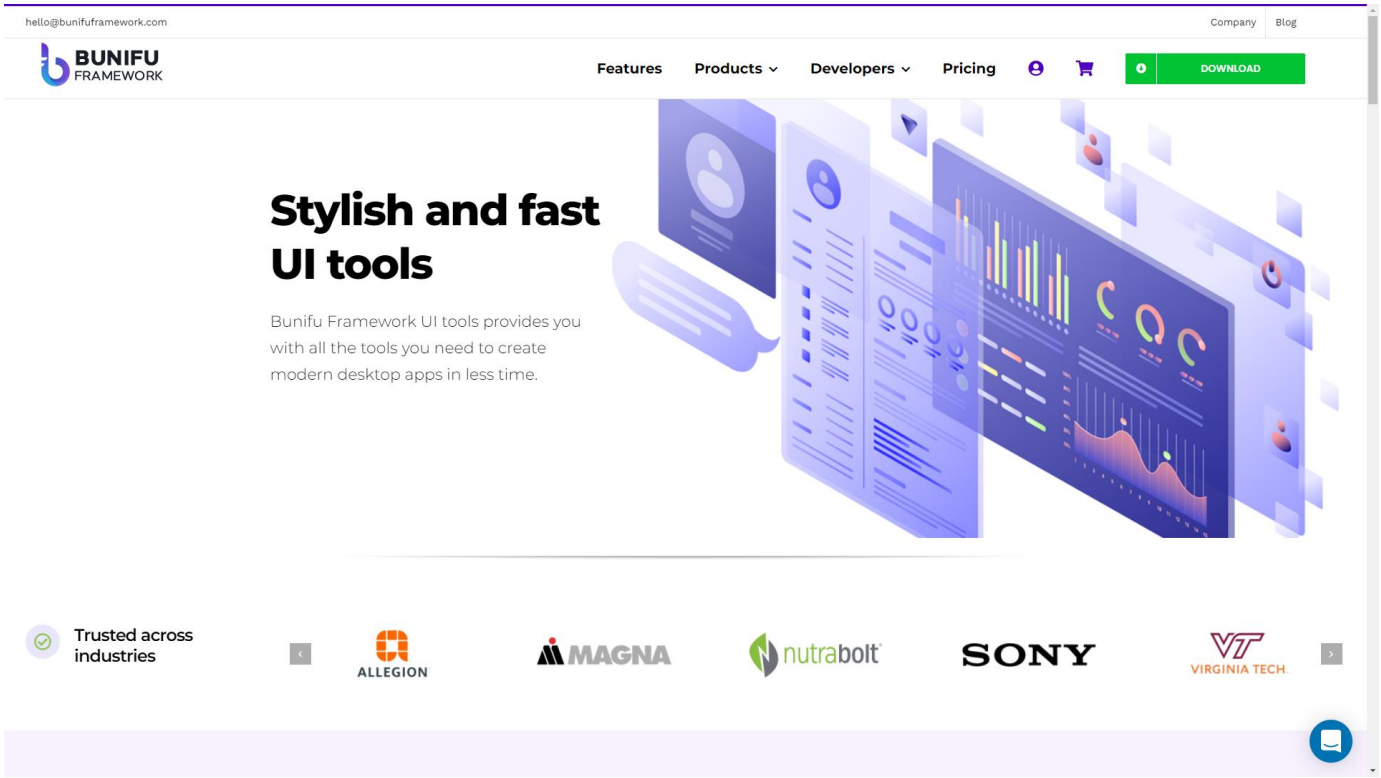

Рис. 1.1. Головна сторінка платформи Bunifu Framework

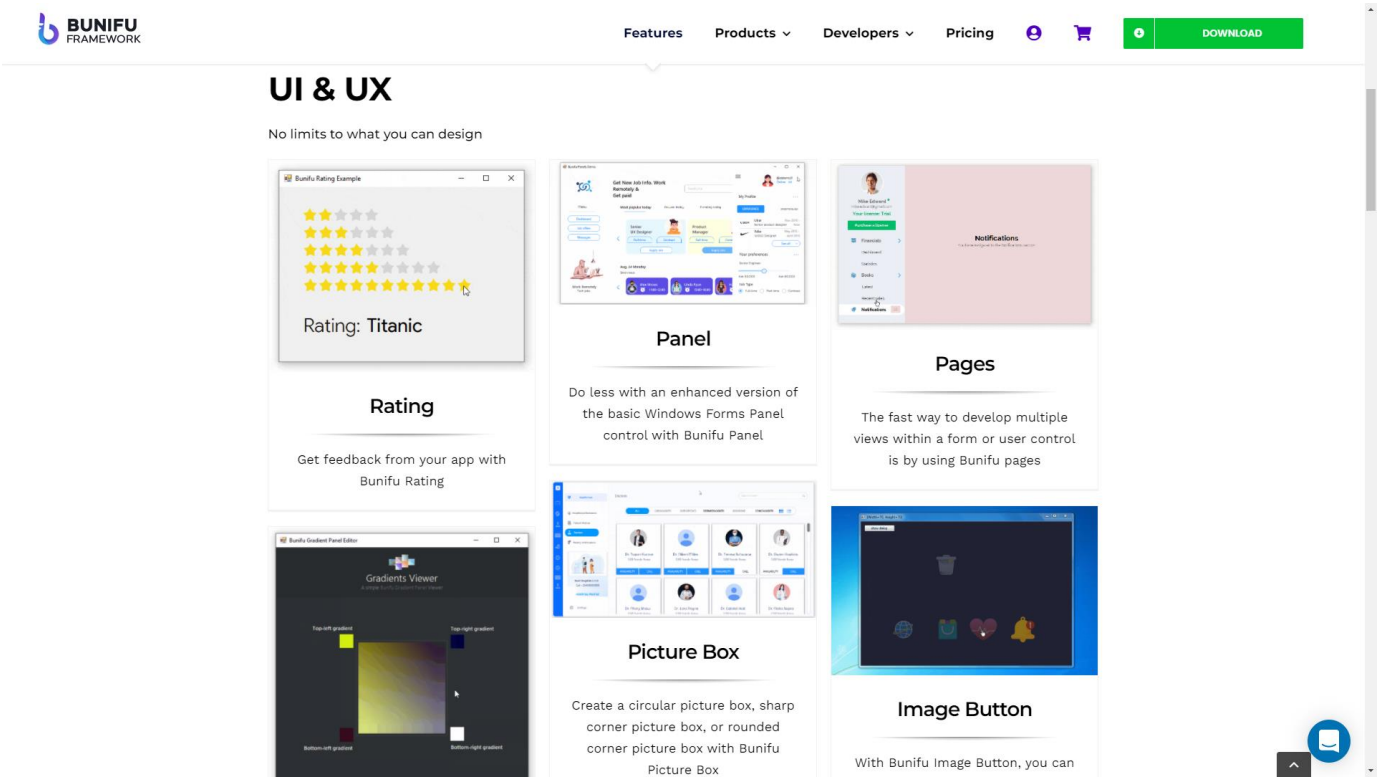

Рис. 1.2. Приклад інтерфейсу користувача на платформі Bunifu Framework

Переваги:

– забезпечення сучасними швидкими елементами керування інтерфейсом користувача. Ідеальні інструменти користувальницького інтерфейсу для WinForms C# і VB.NET розробка додатків;

– простота. Ніяких роздутих властивостей. Саме те, що потрібно для створення приголомшливих додатків WinForms;

– сучасність. Bunifu UI WinForms дозволяють розробникам програмного забезпечення швидко створювати сучасні програми;

– продуктивність. Багато фреймворків сильно впливають на систему. Bunifu UI WinForms слабо навантажує систему.

1.2. Призначення розробки та постановка задачі

Розробка Media Player на C# з використанням Bunifu UI.

Bunifu UI - це керовані інструменти, які допомагають створювати приголомшливі інтерфейси настільних додатків. Це гарантує скорочення часу розробки для Microsoft Visual Studio.NET програмного забезпечення.

Розроблений медіаплеєр буде використовуватися користувачами, які бажають отримати новий EX(User Experience) та гарно провести час. З цим допоможуть хороші фільми і музика - вони здатні відвернути увагу від навколишніх проблем та подарувати користувачеві гарний настрій.

Медіаплеєр призначений для користувачів ПК з операційною системою Windows.

1.3. Підстави для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується з наказом ректора.

Таким чином підставами для розробки (виконанням кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп’ютерні Науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 317-с від 07.06.2021р;
- завдання на кваліфікаційну роботу на тему «Розробка Media Player на C# з використанням Bunifu UI framework».

1.4. Постановка завдання

Головною метою виробничої технологічної практики є розробити Media Player за допомогую інструменту Bunifu UI, який буде дозволяє відтворити аудіо, відео-формати, зберігати плейлисти, зчитувати багато форматів та буде мати унікальний інтерфейс.

Даний Media Player повинен мати вродливий та унікальний інтерфейс, володіти інтуїтивно зрозумілим інтерфейсом і буди простим в експлуатації .

Поставлена мета може бути досягнена при виконані наступних вимог:

– вивчення предметної галузі розв’язуваного завдання;

– проведення порівняльної характеристики можливостей аналогічних застосунків;

– вибір релевантної архітектури програми;

– розробка структур вхідних і вихідних даних для проектованого програмного забезпечення;

– написання програмного коду застосунку;

– розробка рекомендацій щодо застосування програми.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Для коректного запуску додатку потрібно встановити Directx версії 11, на більш ранніх версіях додаток не може бути запущений, через набір використаних функцій у медіаплеєрі.

DirectX - це набір API, розроблених для вирішення завдань, пов'язаних з програмуванням під Microsoft Windows.

Оскільки проект вже запакований у виконувальний файл, від користувача потрібно тільки запустити його.

Також при запуску можливі оповіщення про можливі проблеми:

Warning — нестандартна ситуація з програмою. Не являється критичною, але користувачеві варто приділити їй увагу.

Error — помилка в роботі програми. Деякі функції або файли могли бути недоступні.

Кінцевий продукт повинен дотримуватися наступних функціональних вимог:

- інтуїтивно зрозумілий інтерфейс користувача;
- дані повинні виводитися користувачем на екрані;
- дані можуть зберігатися і видалятися користувачем.

1.5.2. Вимоги до інформаційної безпеки

Оскільки додаток вже запакований, якихось вимог до інформаційної безпеки не потрібно, достатньо просто дотримуватись ієрархії внутрішніх файлів, для уникнення можливих збоїв додатку.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для більш надійного функціювання додатку потрібний DirectX версії 11 та досить сучасне обладнання графічної карти (версія драйверу, яка рекомендована — 398.63). Додаток, спакований на запуск у середі Windows, таким чином додаток не буде запускатись на інших платформах (IOS, Linux, тощо.)

1.5.4. Вимоги до інформаційної та програмної сумісності

Основною мовою програмування було використано C#.

Увесь додаток був розробленій у IDE Microsoft Visual Studio.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення інформаційної системи

Результатом даної кваліфікаційної роботи має бути програма медіалеєр створений за допомогою Bunifu UI framework. Розроблений медіалеєр буде використовуватися користувачами, які бажають відпочити та гарно провести час.

Основне призначення програми:

- Відтворення аудіо форматів;
- Відтворення відео форматів;
- програма призначена для використання на платформі ПК (Windows);
- спроектований додаток забезпечено функціональним інтерфейсом(головне меню).

Медіплеєр підтримує такі формати як:

- mp3;
- wav;
- mp4;
- mov;
- wmv;
- mpg;
- avi;
- 3gp;
- flv;

2.2. Опис застосованих математичних методів

При розробці медіаплеєру у середовищі Visual Studio, було використано такі математичні методи: арифметичні оператори додавання, віднімання, множення, ділення.

Оператор додавання (+) обчислює суму своїх операндів.

Оператор віднімання (-) віднімає правий операнд з лівого.

Оператор множення (*) обчислює добуток операндів.

Оператор ділення (/) ділить лівий операнд на правий.

Оператор інкременту ++ збільшує операнд на 1. Операндом повинна бути змінна, властивість або індексатор.

Оператор інкременту підтримується в двох формах: Постфіксний оператор інкременту (x++) і префіксний оператор інкременту (++x).

Унарний оператор декременту зменшує операнд на 1. Операндом повинна бути змінна, властивість або індексатор.

Оператор декременту підтримується у двох формах: Постфіксний оператор декременту (x--) і префіксний оператор декременту (--x).

Унарний оператор + повертає значення отриманого операнда. Унарний оператор-змінює знак операнда на протилежний.

2.3. Опис використаної архітектури та шаблонів проектування

При розробці мультіплеєру, було обрано класичний шаблон програмування Model View Presenter (MVP) - це шаблон, який вперше з'явився в IBM, а потім використовувався в Taligent в 1990-х. MVP є похідним від MVC, при цьому має дещо інший підхід. У MVP представлення не тісно пов'язане з моделлю, як це було в MVC.

Model View Presenter зайняв місце контролера і відповідає за переміщення даних, введених користувачем, а також за оновлення подання при змінах, які відбуваються в моделі. Presenter спілкується з поданням через інтерфейс, який

дозволяє збільшити тестованість, так як модель може бути замінена на спеціальний макет для модульних тестів.

При розробці медіалеєру було використано - Microsoft Visual Studio 2019. Завдяки вікну Оглядач рішень, можна зручно переглянути структуру проекту на мові програмування C#. За допомогою функціоналу Visual Studio 2019, файли проекту зручно збираються у Рішення (solution). Рішення являє собою структуру для організації проектів в Visual Studio.

Завдяки цьому проект має наступну архітектуру(рис. 2.1):

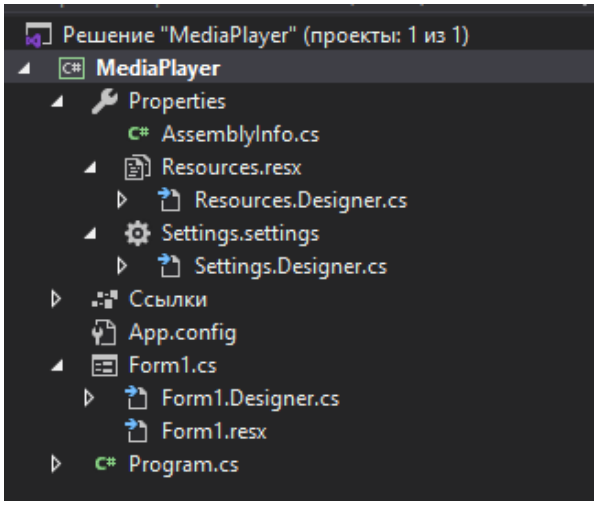

Рис. 2.1. Структура проекту у Visual Studio 2019

2.4. Опис використаних технологій та мов програмування

В якості мови програмування було використано C#. Це сучасна об'єктно-орієнтована і типобезпечна мова програмування. Дозволяє розробникам створювати безліч типів безпечних і надійних додатків, що працюють в екосистемі .NET. C# відноситься до широко відомого сімейства мов C, і здасться добре знайомим будь-кому, хто працював з C, C++, Java або JavaScript версій.

Це об'єктно і компонентно-орієнтована мова програмування. C# надає мовні конструкції для безпосередньої підтримки такої концепції роботи.

Завдяки цьому C # підходить для створення і застосування програмних компонентів. З моменту створення мова C# збагатилася функціями для підтримки нових робочих навантажень і сучасними рекомендаціями з розробки ПЗ.

Програми C# виконуються в .NET, віртуальній системі виконання, що викликає загальномовне середовище виконання (CLR) і набір бібліотек класів. Середовище CLR - це реалізація загальномовної інфраструктури мови (CLI), що є міжнародним стандартом, від корпорації Майкрософт. CLI є основою для створення середовищ виконання та розробки, в яких мови та бібліотеки прозоро працюють один з одним.

Вихідний код, написаний на мові C# компілюється в проміжну мову (IL), який відповідає специфікаціям CLI. Код на мові IL і ресурси, в тому числі растрові зображення і рядки, зберігаються в збірці, зазвичай з розширенням .dll. Збірка містить маніфест з інформацією про типи, версії, мовою і регіональних параметрах для цієї збірки.

При виконанні програми C# збірка завантажується в середу CLR. Середовище CLR виконує JIT-компіляцію з коду на мові IL в інструкції машинного мови. Середовище CLR також виконує інші операції, наприклад, автоматичне збирання сміття, обробку виключень та управління ресурсами. Код, що виконується середовищем CLR, іноді називають «керованим кодом», щоб підкреслити відмінності цього підходу від «некерованого коду», який відразу компілюється в машинну мову для певної платформи.

Забезпечення взаємодії між мовами є ключовою особливістю .NET. Код IL, створений компілятором C#, відповідає специфікації загальних типів (CTS). Код IL, створений з коду на C#, може взаємодіяти з кодом, створеним з версій. NET для мов F#, Visual Basic, C++ і будь-яких інших з більш ніж 20 мов, сумісних з CTS. Одна збірка може містити кілька модулів, написаних на різних мовах .NET, і всі типи можуть посилатися один на одного, як якщо б вони були написані на одній мові.

Windows Forms дозволяє розробляти інтелектуальні клієнти. Інтелектуальний клієнт - це програма з повнофункціональним графічним інтерфейсом, просте в розгортанні і оновленні, здатне працювати при наявності або відсутності підключення до Інтернету і використовує більш безпечний доступ до ресурсів на локальному комп'ютері в порівнянні з традиційними додатками Windows.

Windows Forms - це технологія інтелектуальних клієнтів для NET Framework. Вона являє собою набір керованих бібліотек, що спрощують виконання стандартних завдань, таких як читання з файлової системи і запис в неї. За допомогою такого середовища розробки, як Visual Studio, можна створювати інтелектуальні клієнтські програми Windows Forms, які відображають інформацію, запитують введення від користувачів і обмінюються даними з віддаленими комп'ютерами по мережі.

У Windows Forms форма - це візуальна поверхня, на якій відображаються відомості для користувача. Зазвичай додаток Windows Forms будується шляхом приміщення елементів управління на форму і написання коду для реагування на дії користувача, такі як клацання миші або натискання клавіш. Елемент керування - це окремий елемент інтерфейсу користувача, призначений для відображення або введення даних.

Windows Forms включає широкий набір елементів управління, які можна додавати на форми: текстові поля, кнопки, розкривні списки, перемикачі і навіть веб-сторінки. Список всіх елементів управління, які можна використовувати у формі, представлені в розділі елементи управління для використання у формах Windows Forms. Якщо існуючий елемент керування не відповідає потребам, У Windows Forms можна створити власні елементи керування за допомогою класу UserControl.

До складу Windows Forms входять багатофункціональні елементи інтерфейсу, що дозволяють відтворювати можливості таких складних додатків, як Microsoft Office. Використовуючи елементи керування ToolStrip і MenuStrip,

можна створювати панелі інструментів і меню, що містять текст і малюнки, підменю та інші елементи управління, такі як текстові поля і поля зі списками.

За допомогою конструктор Windows Forms перетягування в Visual Studio можна легко створювати Windows Forms додатки. Досить виділити елемент управління курсором і помістити його в потрібне місце на формі. Для подолання труднощів, пов'язаних з вирівнюванням елементів управління, конструктор надає такі засоби, як лінії сітки і лінії прив'язки. А також при використанні Visual Studio або компіляції з командного рядка можна використовувати FlowLayoutPanel, TableLayoutPanel, SplitContainer елементи управління, і для створення розширених макетів форм за менший час.

При виконанні користувачем будь-якої дії з формою або одним з її елементів управління створюється подія. Додаток реагує на ці події за допомогою коду і обробляє події при їх виникненні.

У багатьох додатках потрібно відображати дані з бази даних, XML-файлу, веб-служби XML або іншого джерела даних. Windows Forms надає гнучкий елемент управління з ім'ям DataGridView для відображення таких табличних даних в традиційному форматі рядків і стовпців так, що кожен фрагмент даних займає свою власну комірку. За допомогою DataGridView можна, крім іншого, налаштувати зовнішній вигляд окремих осередків, зафіксувати рядки і стовпці на своєму місці, а також забезпечити відображення складних елементів управління всередині осередків.

При використанні інтелектуальних клієнтів Windows Forms можна легко підключатися до джерел даних по мережі. BindingSource компонент представляє з'єднання з джерелом даних і надає методи для прив'язки даних до елементів управління, переходу до попередніх і наступних записів, зміни записів і збереження змін у вихідному джерелі. Елемент керування BindingNavigator надає простий інтерфейс на основі компонента BindingSource для переходу між записами.

Можна легко створювати елементи керування з прив'язкою до даних за допомогою вікна «джерела даних». У ньому наводяться наявні в проекті джерела даних, такі як бази даних, веб-служби та об'єкти. Створювати елементи управління з прив'язкою до даних можна шляхом перетягування об'єктів з цього вікна в форми проекту. Також можна пов'язувати існуючі елементи управління з даними, перетягуючи об'єкти з вікна «джерела даних» в існуючі елементи управління.

Інший тип прив'язки даних, який можна керувати в Windows Forms - це Параметри. Більшість інтелектуальних клієнтських додатків повинні зберігати деякі відомості про свій стан під час виконання, такі як останні відомі розміри форм, а також зберігати користувацькі переваги, наприклад місце збереження файлів за замовчуванням. Параметри програми відповідає цим вимогам, надаючи простий спосіб зберігання обох типів відомостей на клієнтському комп'ютері. Після визначення цих параметрів за допомогою Visual Studio або редактора коду параметри зберігаються у форматі XML і автоматично зчитуються назад в пам'ять під час виконання.

Система Microsoft Visual Studio є найскладнішою інтегрованим середовищем розробки (integrated development enviroment - IDE), доступною для програмістів в даний час. Вона є результатом довгої історії розвитку мов програмування та інтерфейсів і увібрала в себе досягнення багатьох середовищ розробки програмного забезпечення(рис 2.2).

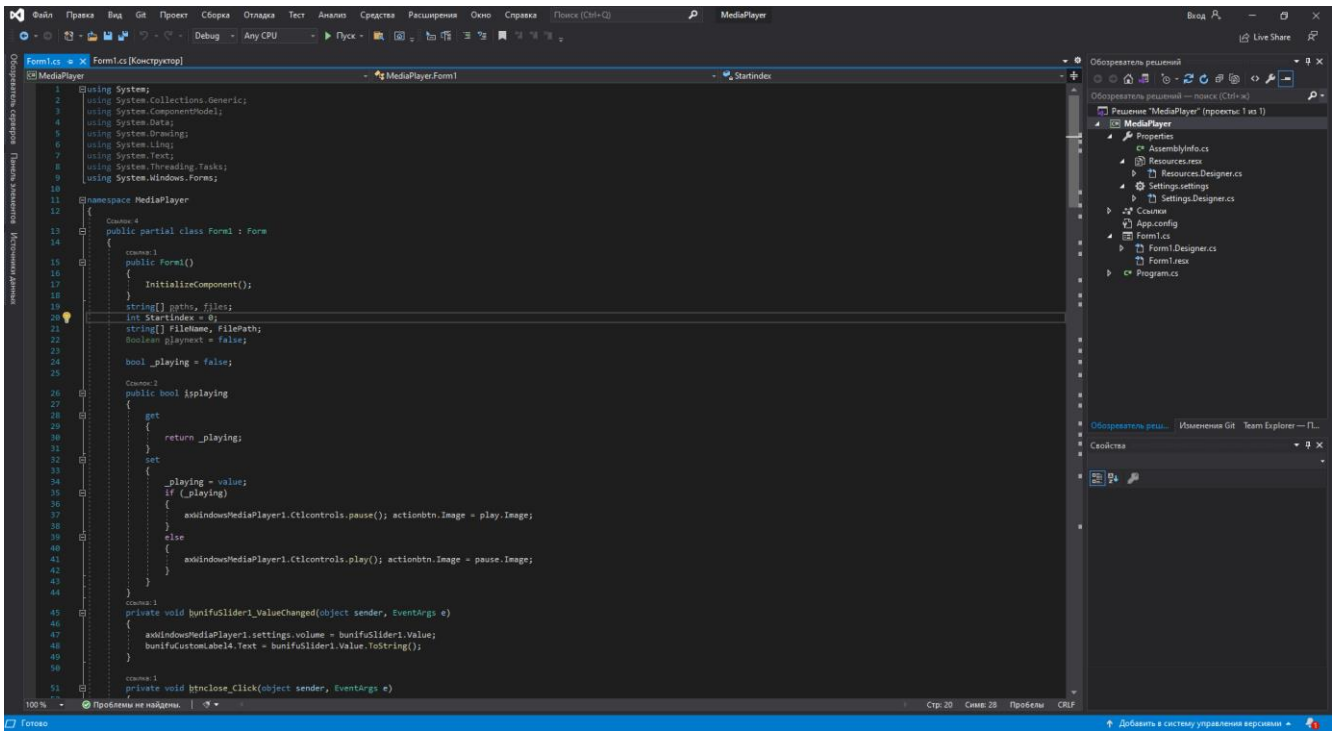

Рис. 2.2. Інтерфейс середовища Visual Studio

Функціональність Visual Studio охоплює всі етапи розробки програмного забезпечення, надаючи сучасні інструменти для написання коду, проектування інтерфейсів, складання, налагодження та тестування додатків. Можливості Visual Studio можуть бути доповнені шляхом підключення необхідних розширень.

Редактор коду Visual Studio підтримує підсвічування синтаксису, вставку фрагментів коду, відображення структури і пов'язаних функцій. Істотно прискорити роботу допомагає технологія IntelliSense - автозавершення коду в міру введення.

Вбудований відладчик Visual Studio використовується для пошуку і виправлення помилок у вихідному коді, в тому числі на низькому апаратному рівні. Інструменти Діагностики дозволяють оцінити якість коду з точки зору продуктивності і використання пам'яті.

Дизайнер форм Visual Studio незамінний при розробці програм з складним інтерфейсом, допомагаючи спроектувати зовнішній вигляд майбутнього додатка і роботу кожного елемента інтерфейсу.

Visual Studio надає комплекс інструментів для автоматизації тестування додатків в частині перевірки роботи інтерфейсів, модульного і навантажувального тестування.

Для командних проектів Visual Studio пропонує підтримку групової роботи, дозволяючи виконувати спільне редагування і налагодження будь-якої частини коду в реальному часі, а в якості системи управління версіями використовувати Team Foundation або Git.

Основним розширенням файлу, асоційованим з Microsoft Visual Studio, є SLN-Visual Studio Solution File (Файл рішення Visual Studio), при відкритті якого в програму завантажуються всі дані і проекти, пов'язані з розроблюваним програмним рішенням.

При створенні в Visual Studio програми або веб-сайту все починається з проекту. З логічної точки зору проект містить всі файли, які будуть скомпільовані в виконувану програму, бібліотеку або веб-сайт. Сюди входять файли з вихідним кодом, значками, зображеннями, даними і т. д. Проект також містить параметри компілятора та інші файли конфігурації, які можуть знадобитися різним службам або компонентам, з якими взаємодіє програма.

Visual Studio використовує MSBuild для створення кожного проекту в рішенні, і кожен проект містить файл проекту MSBuild. Розширення імені файлу відображає тип проекту, наприклад проект C #(CSPROJ), проект Visual Basic (VBPROJ) або проект бази даних (DBPROJ). Файл проекту являє собою XML-документ, який містить всі відомості та інструкції, необхідні MSBuild для складання проекту, включаючи вміст, вимоги до платформи, відомості про управління версіями, параметри веб-сервера або сервера баз даних, а також виконувані завдання.

Файли проекту засновані на схемі XML MSBuild. Щоб переглянути вміст файлів проекту в новому стилі SDK в Visual Studio, клацніть правою кнопкою миші вузол проекту в браузері рішень і виберіть пункт Змінити <projectname> . Щоб переглянути вміст проекту. NET Framework або іншого проекту в цьому стилі, спочатку треба вивантажить проект. Потім треба клацнути на проект правою кнопкою миші і виберати пункт Змінити projectname

В центрі інтерфейсу розташована вкладка Конструктор (рис. 2.3). Саме тут можна побачити, як буде виглядати медіаплеєр, і розташовувані об’єкти на формі. Слід розуміти, що все те, що відображається на цій вкладці відрізнятиметься від того, що буде відображатися в процесі відладки.

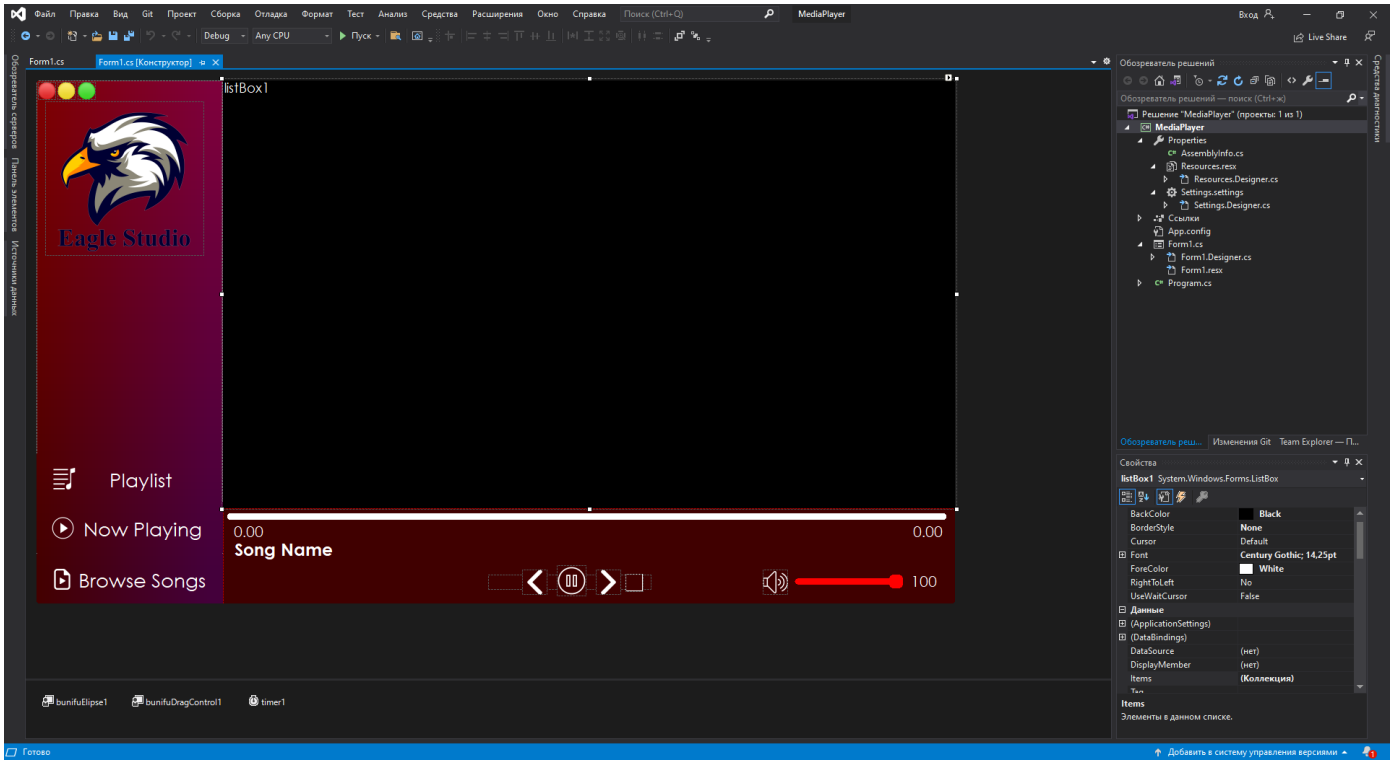

Рис. 2.3. вкладка Конструктор

Вкладка Панель Елементів (рисунок 2.4) містить список всіх присутніх на сцені об’єктів у вигляді дерева, гілки якого вкладені одна в одну відповідно до ієрархічних зв’язків між об’єктами. Вона надає можливість виділення об’єктів

по іменах, позбавляючи від необхідності пошуку об’єкта на сцені. Ієрархічні зв’язки поєднують об’єкти один з одним.

Багато команд меню доступні в контекстному меню різних елементів в браузері рішень. До таких команд відноситься збірка проекту, управління пакетами NuGet, додавання посилань, перейменування файлу і запуск тестів.

Одна з основних переваг над іншими інтегрованими середовищами розробки полягає в тому, що компанія Microsoft пропонує користувачам реальну можливість переміщати елементи при проектуванні як веб, так і складних клієнтських додатків. Всі ці елементи розміщені у вікні Панель Елементів, доступ до якого можна отримати за допомогою меню Вид.

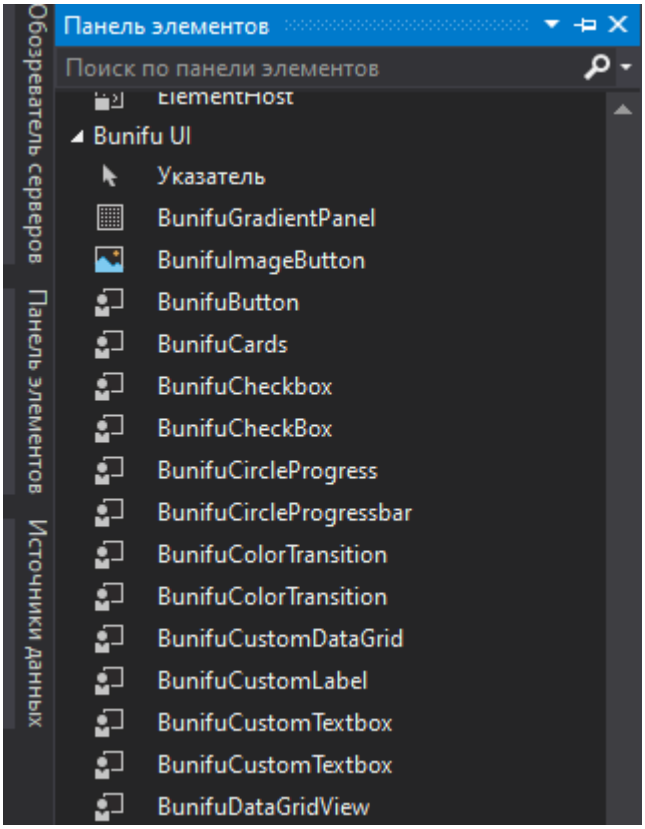

Рис. 2.4. Частина редактора, яка демонструє тільки вкладку Панель Елементів

Вікно Панель Елементів має цікаву властивість: можна скопіювати фрагмент коду в нього, клацнувши на області коду і перетягнувши її в вікно Панель Елементів. Можна також перейменувати і переупорядкувати фрагменти коду, зробивши їх дійсно корисними для презентацій або зберігання часто використовуваних фрагментів.

Вікно Панель Елементів містить всі доступні компоненти для активного в даний момент документа, відкритого в головному робочому вікні. Ними можуть бути візуальні компоненти, наприклад кнопки і текстові поля; невидимі, Сервісні об'єкти, наприклад таймери або реєстраційні журнали системних подій; і навіть елементи дизайну, такі як класи і інтерфейсні об'єкти, використовувані в інструменті Class Designer.

Система Visual Studio 2019 групує доступні компоненти, а не змішує їх в одну купу. Таке угруповання за замовчуванням дозволяє простіше знаходити необхідні елементи контролю; наприклад, компоненти для роботи з даними містяться в окремій групі Data.

Незалежно від представлення компонентів спосіб їх використання в програмі залишився колишнім: треба клацнути і перетягнути бажаний компонент на форму активного документа або двічі клацнути на компоненті в системі Visual Studio, щоб автоматично додати його екземпляр. Візуальні компоненти, такі як кнопки і текстові поля, з'являються на формі, після чого користувач може переміщати їх, змінювати розміри або фіксувати за допомогою властивостей сітки. Невізуальні компоненти, такі як таймер, відображаються у вигляді піктограм з асоційованими мітками в області форми, призначеної для невидимих компонентів.

У верхньому лівому куті показана група Gettingstarted Controls з єдиним компонентом SampleButton. По суті, рядок «GettingStarted» - це ім'я проекту WPF. Він містить елемент керування SampleButton. Коли ви починаєте створювати свої власні компоненти або Елементи управління, то, замість того, щоб надати вручну створювати нову закладку і повторювати весь процес

додавання кожного елемента, система Visual Studio 2019 автоматично переглядає всі проекти вашого рішення. Як тільки компоненти або Елементи управління будуть ідентифіковані, в проекті буде створена нова закладка, на яку будуть додані відповідні елементи з піктограмами та іменами класів, заданими за замовчуванням, в даному випадку SampleButton. Коли використовується компонент, в області невидимих елементів з'являється відповідна піктограма.

Іноді виявляється, що конкретний компонент, який потрібен, відсутній у списку Toolbox. Більшість основних компонентів. Net в цьому списку є, але деяких немає. Наприклад, компонент WebClient class у списку Toolbox за замовчуванням не вказано. Керовані програми можуть також використовувати компоненти COM. Після додавання у вікно Toolbox об'єкти COM можуть бути використані точно так само, як звичайні компоненти .NET, і якщо вони закодовані правильно, то можна програмувати роботу з ними як зазвичай, використовуючи вікно Properties і посилаючись на їх методи, властивості і події в своєму коді.

Для того щоб додати компонент в своє вікно Toolbox, потрібно клацнути правою кнопкою миші на групі компонентів і виконати команду Choose Items. Через деякий час (на повільному комп'ютері цей процес може зайняти кілька секунд, тому що машина повинна переглянути кеш .Net, щоб визначити всі можливі компоненти, які можна вибрати) відобразиться список. NET Framework components(рис.2.5).

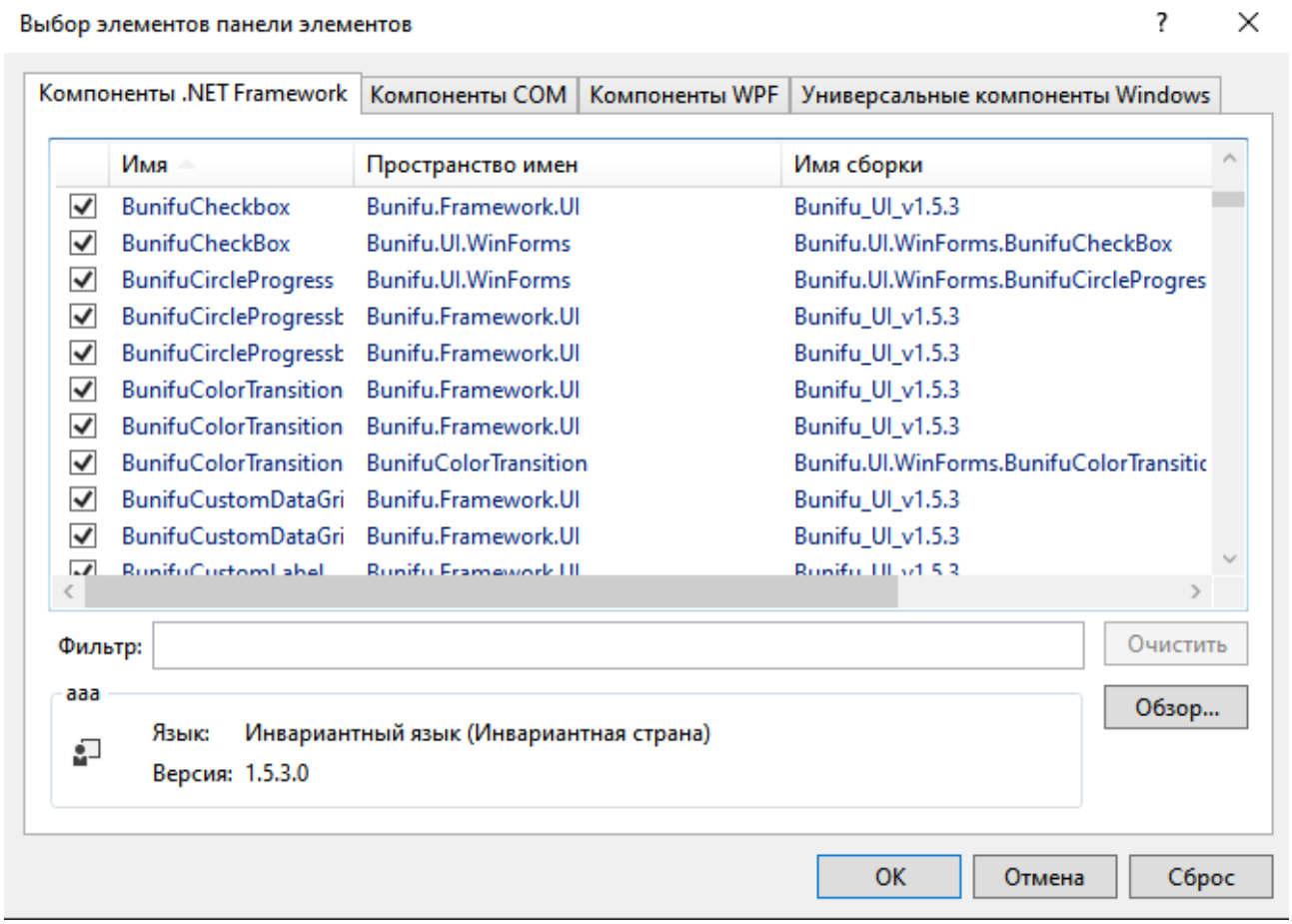

Рис. 2.5. Вибір елементів панелі елементів

На панелі інструментів у верхній частині владки Оглядач рішень (рис.2.6) є кнопки для перемикання з подання рішення в представлення папки, фільтрації очікують змін, відображення всіх файлів, згортання всіх вузлів, перегляду сторінок властивостей, перегляду коду в редакторі коду і т. д.

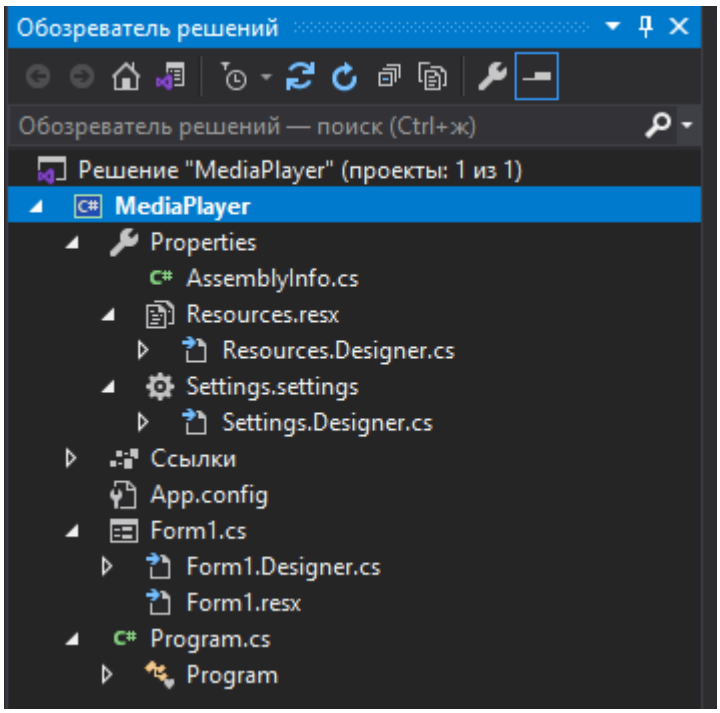

Рис. 2.6. Частина редактора, яка демонструє тільки вкладку Оглядач рішень

Інструментальне вікно Оглядач рішень забезпечує зручне візуальне представлення рішення, проектів і елементів.

З кожним проектом пов'язана окрема піктограма, яка, як правило, вказує тип проекту і мову, на якому він написаний. У цього правила є кілька винятків, наприклад, проекти установки не мають мови програмування, на якому вони створюються.

Один з вузлів звертає на себе особливу увагу, оскільки він виділений напівжирним шрифтом. Це означає, що цей проект є стартовим, інакше кажучи, проектом, який запускається, коли виконується команда Debug --> Start Debugging. Стартовими можна зробити відразу кілька проектів. Для цього слід використовувати діалогове вікно Solution Properties.

Панель інструментів у верхній частині вікна Оглядач рішень дозволяє налаштувати зовнішній вигляд цього вікна, а також задати комбінації клавіш для різного зовнішнього вигляду окремих елементів. Наприклад, кнопка

Показати всі файли - відкриває лістинг рішення і дозволяє показати на екрані додаткові файли і папки(рис.2.7).

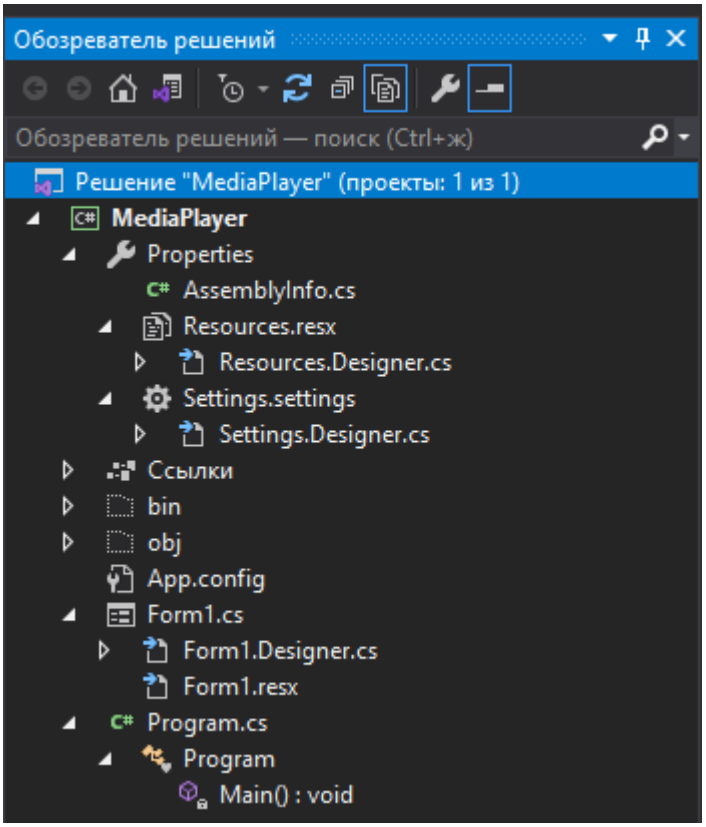

Рис. 2.7. Додаткові файли і папки

У цьому розгорнутому представленні можна побачити всі файли і папки, що відносяться до структури проекту. На жаль, при зміні файлової системи вікно Оглядач рішень не оновлюється автоматично, щоб відобразити ці зміни. Для того щоб забезпечити правильне представлення списку файлів і папок, можна використовувати кнопку Оновити.

Інструментальна панель Оглядач рішень чутлива до контексту, тому в залежності від обраного типу вузла на ній відображаються різні кнопки.

До основних дій, виконуваних у вікні Оглядач рішень, відносяться додавання, видалення і перейменування проектів і елементів. Для того щоб додати новий проект в існуюче рішення, слід виконати команду Добавить --> Новий проект в контекстному меню, пов'язаному з вузлом Рішення.(рис.2.8). У цьому випадку відкриється діалогове вікно, показане на малюнку нижче, яке з невеликими змінами успадковано від попередньої версії системи Visual Studio. Тепер шаблони проектів можна сортувати і шукати.

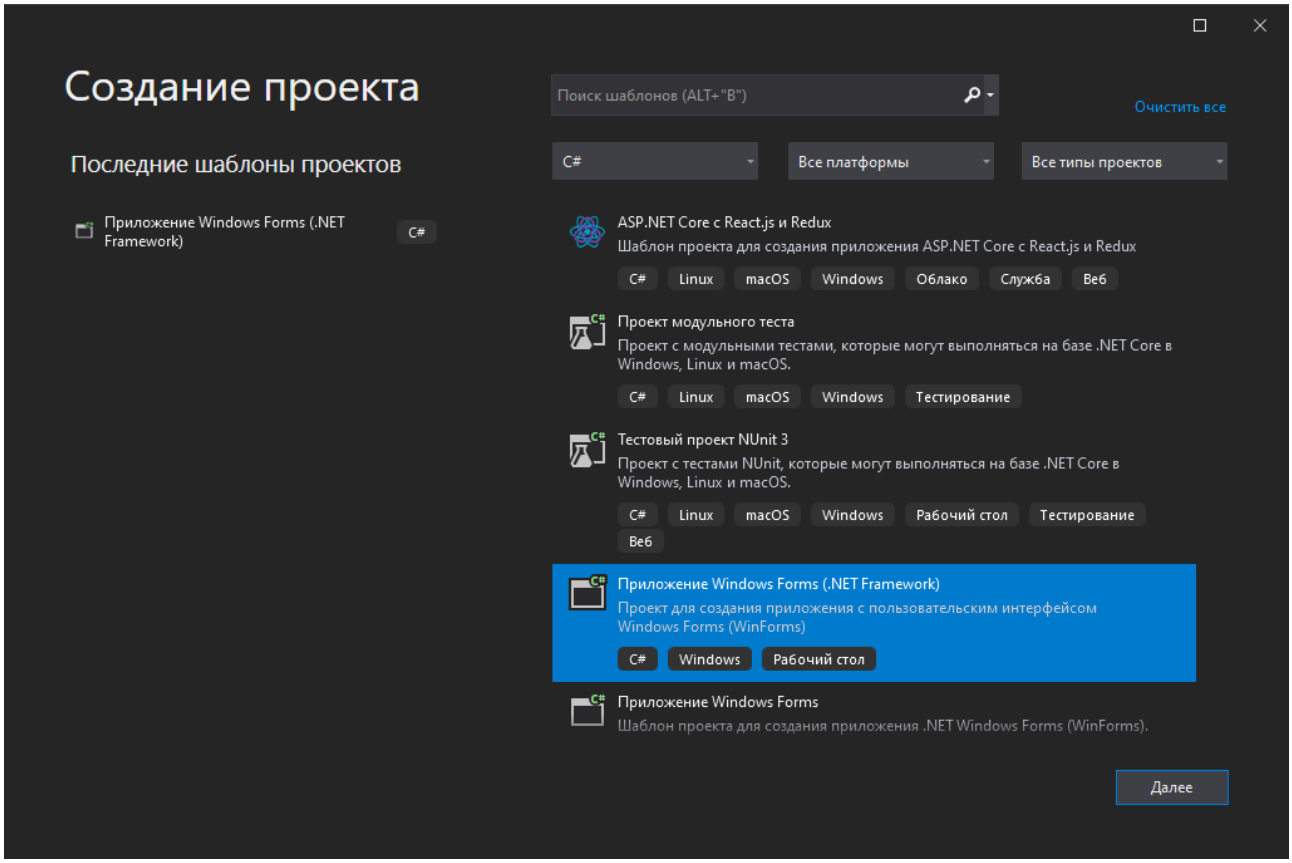

Рис. 2.8. Вікно створення проекта

Крім того, слід виділити ще одну особливість цього діалогового вікна - можливість вибирати різні версії платформи. Якщо ви працюєте зі старим проектом і не хочете переводити його в нову версію платформи .NET Framework, то все одно можете скористатися новими можливостями, такими як поліпшена технологія IntelliSense. В іншому випадку довелося б інсталювати як

систему Visual Studio 2019, так і попередню версію, щоб створювати проекти для старіших версій платформи. Вибір платформи також включений в критерії сортування, що обмежує список доступних шаблонів проекту лише тими шаблонами, які сумісні з обраною версією платформи .NET Framework.

2.5. Опис структури програми та алгоритмів її функціонування

Розроблений проект має кореневу структуру, зображену на рисунку 2.11. Лістинг коду міститься у Додатку А.

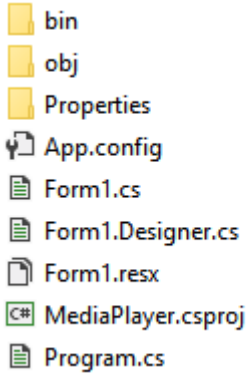

Рис. 2.11. Структура проекту

Головні папки проекту:

- bin – містить двійкові файли, які є фактичним виконуваним кодом для програми або бібліотеки;
- obj – містить об'єктні або проміжні файли, які являють собою скомпільовані двійкові файли, які ще не були пов'язані. Це фрагменти, які будуть об'єднані для створення остаточного виконуваного файлу. Компілятор генерує один об'єктний файл для кожного вихідного файлу, і ці файли поміщаються в папку obj.

Кожна з цих папок далі поділяється на папки Debug і Release , які просто відповідають конфігураціям збірки проекту. Два типи файлів, описаних вище,

поміщаються у відповідну папку в залежності від того, який тип збірки виконується. Це дозволяє легко визначити, які виконувані файли побудовані з налагоджувальними символами, а які були побудовані з включеною оптимізацією і готові до випуску.

Медіаплеєр забезпечено головним меню користувача, за допомогою якого він має можливість зручно обрати необхідний пункт для функціонування програми певним чином. Результат проектування показаний на вкладці конструктора, на якій розташовані компоненти головного меню (рисунок 2.12).

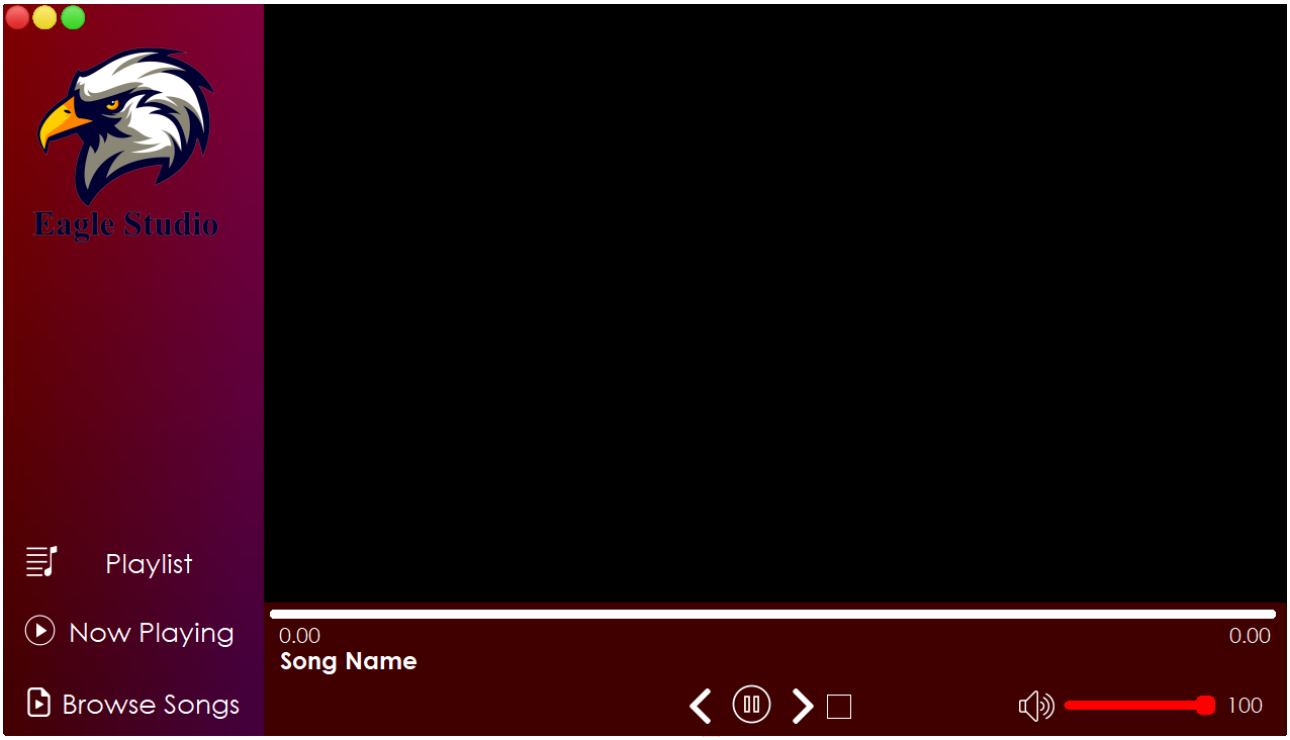

Рис. 2.12. Головне меню плеєра

За завантаження аудіо та відео файлів відповідає елемент BunifuFlatButton, кнопка «Browse Songs»(рис.2.13).

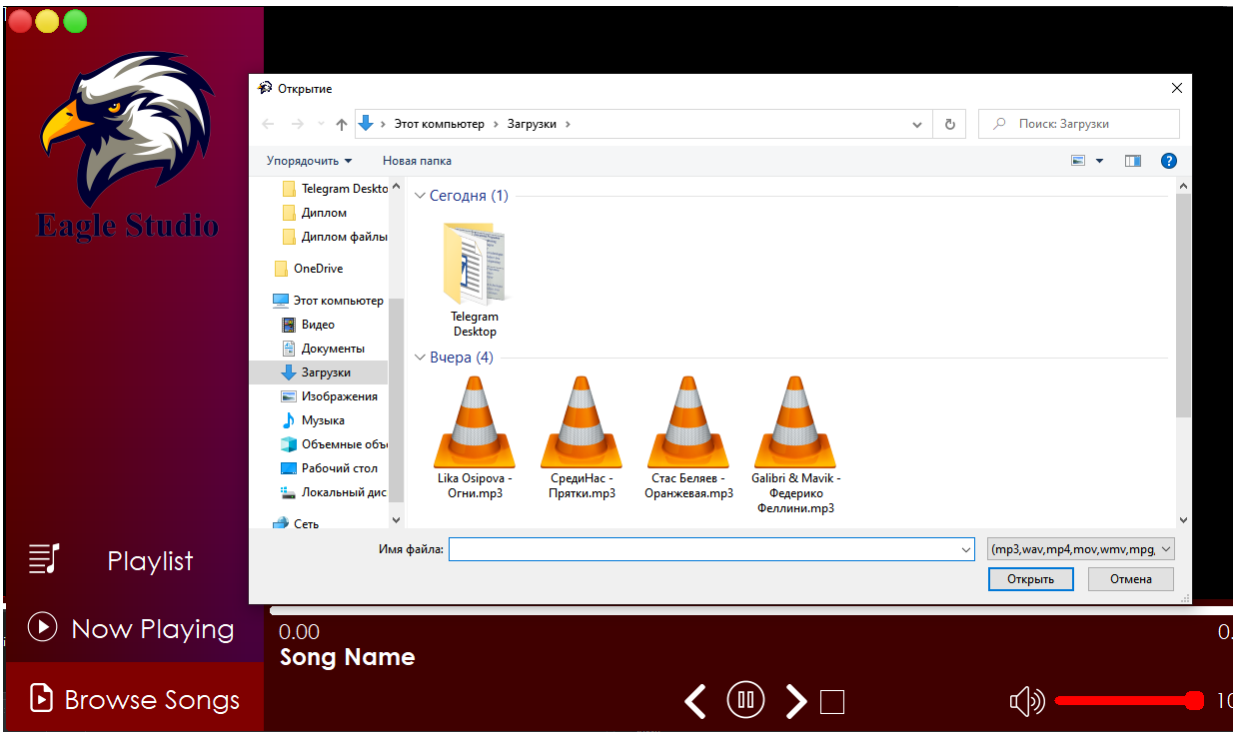

Рис. 2.13. Результат натискання на кнопку Browse Songs

На кнопці Now Playing видно який аудіофайл відтворений в даний момент і також відтворює графіку музики(рис.2.14).

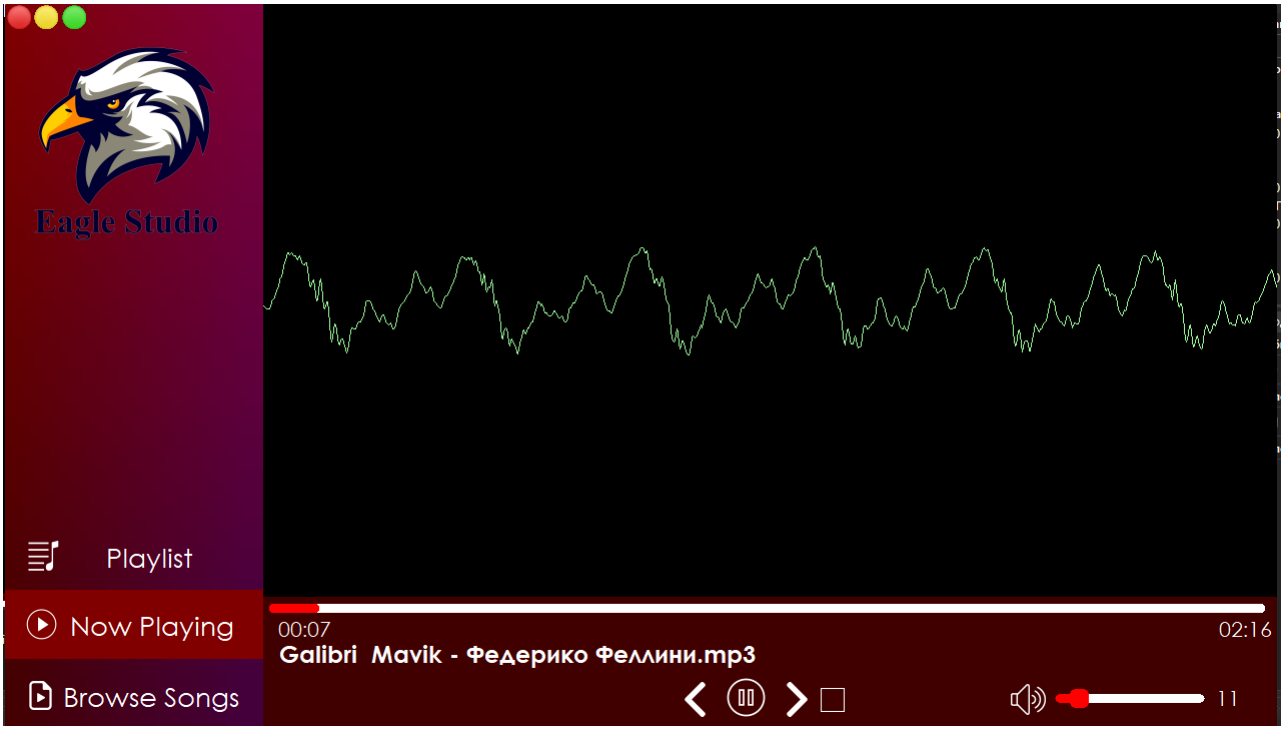

Рис. 2.14. Результат натискання на кнопку Now Playing

Елемент BunifuSlider дозволяє регулювати гучність аудіо і відео файлів(рис. 2.15).

Рис. 2.15. Перемикач гучності

Кнопка Playlist показує перелік завантажених аудіо і відео файлів(рис.2.16).

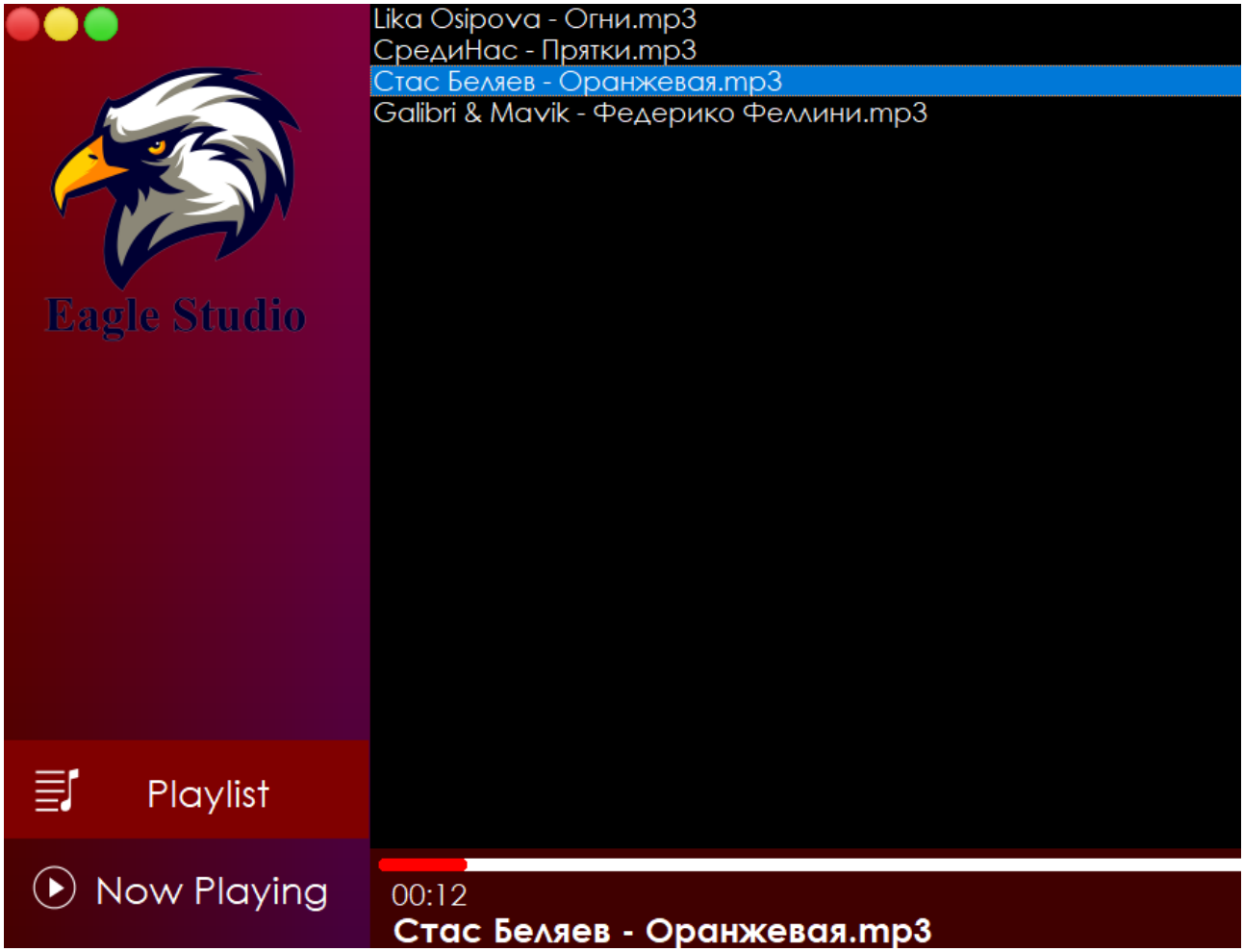

Рис. 2.16. Результат натискання на кнопку Playlist

2.6. Обґрунтування та організація вхідних та вихідних даних програми

Вхідними даними мультиплеєра є аудіо та відео файли, які завантажує мультиплеер.

Вихідними даними є звук, який виводиться на динаміки і зображення на екрані.

2.7. Опис розробленого програмного продукту

2.7.1. Використані технічні засоби

При розробці та тестуванні системи була використана клієнтська персональна ЕОМ з наступними мінімальними характеристиками:

- процесор класу Intel i3;
- монітор 60 гц;
- не менше 2000Мб ОЗУ;
- 20 Гб вільного місяця для середи Visual Studio;
- 59 МБ вільного місця для програми медіаплеєр;
- клавіатура;
- миша.

2.7.2. Використані програмні засоби

Для роботи програмного засобу необхідні такі програмні засоби - Visual Studio версії не менш, ніж 17.

2.7.3. Виклик та завантаження інформаційної системи

Для роботи з розробленим медіаплеєром, необхідно запустити файл з назвою «MediaPlayer.exe» з папки Debug(рис.2.21).

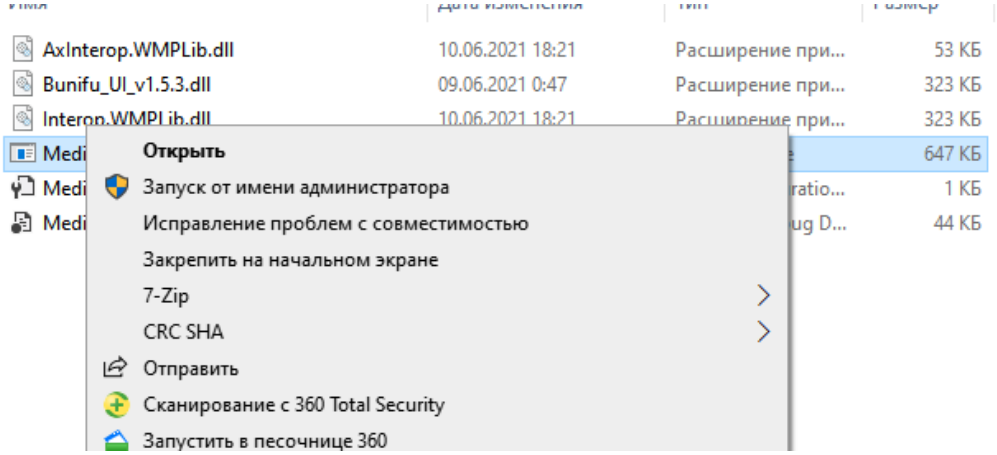

Рис. 2.21. Запуск програми із папки

Після запуску файлу MediaPlayer, користувач побачить інтерфейс медіаплеєру (рис.2.22)

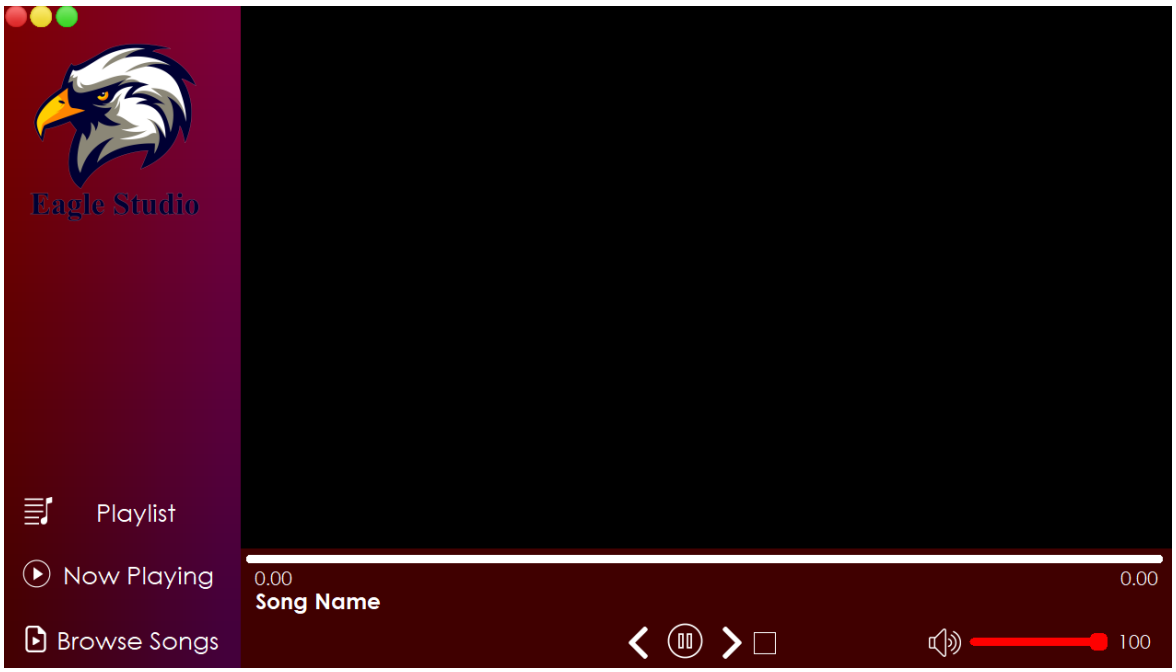

Рис. 2.22. Екран завантаження медіаплеєру

2.7.4. Опис інтерфейсу користувача

При розробці медіалеєру, було спроектовано та розроблено функціональний інтерфейс користувача. Для коректної роботи застосунку, необхідно дочекатися вдалого налаштування та запуску самої програми.

При запуску програми, буде запущено мультіплеєр із розробленим інтерфейсом користувача, завдяки натисканню на елементи якого, користувач зможе обирати медіафайли. Головне меню (рис.2.23) складається із таких елементів графічного інтерфейсу користувача:

- Надпис «Playlist», цей елемент служить для відображення переліку медіафайлів;
- Надпис «Now Playing», елемент відображає аудіо або відеоряд, який відтворюється в реальному часі та його графік;
- Надпис «Browse Songs», дозволяє вибирати і завантажувати файли

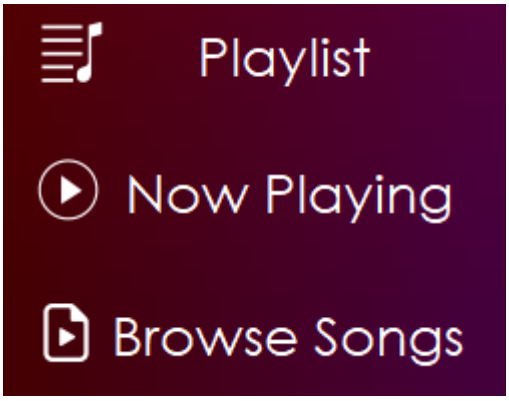

Рис. 2.23. Інтерфейс користувача

Посередині внизу медіаплеєра розташовані кнопки управління перемикання між файлами. Іконка стрілочки вліво дозволяє переграти аудіо, відео файл заново або включити попередній файл. Іконка стрілочки вправо дозволяє відтворити наступний файл. Кругла кнопка між ними дозволяє зупиняти(рис.2.23) і назад відтворювати аудіо, відео файл.(рис.2.24). Іконка

квадрата дозволяє зупинити файл і при натисканні кнопки Play переграти його спочатку. Отже ми маємо зручний і гнучкий інтерфейс перемикання між файлами.

Рис. 2.23. Інтерфейс користувача файл зупинено

Рис. 2.24. Інтерфейс користувача файл відтворено

Також в лівій частині інтерфейсу ми маємо 3 кнопки(рис.2.25):

- Кнопка червоного кольору відповідає за закриття програми;
- Кнопка жовтого кольору відповідає за розгортання і згортання програми на весь екран(рис.2.26);
- Кнопка зеленого кольору відповідає за згортання та розгортання програми.

Рис. 2.25. Інтерфейс користувача кнопки навігації

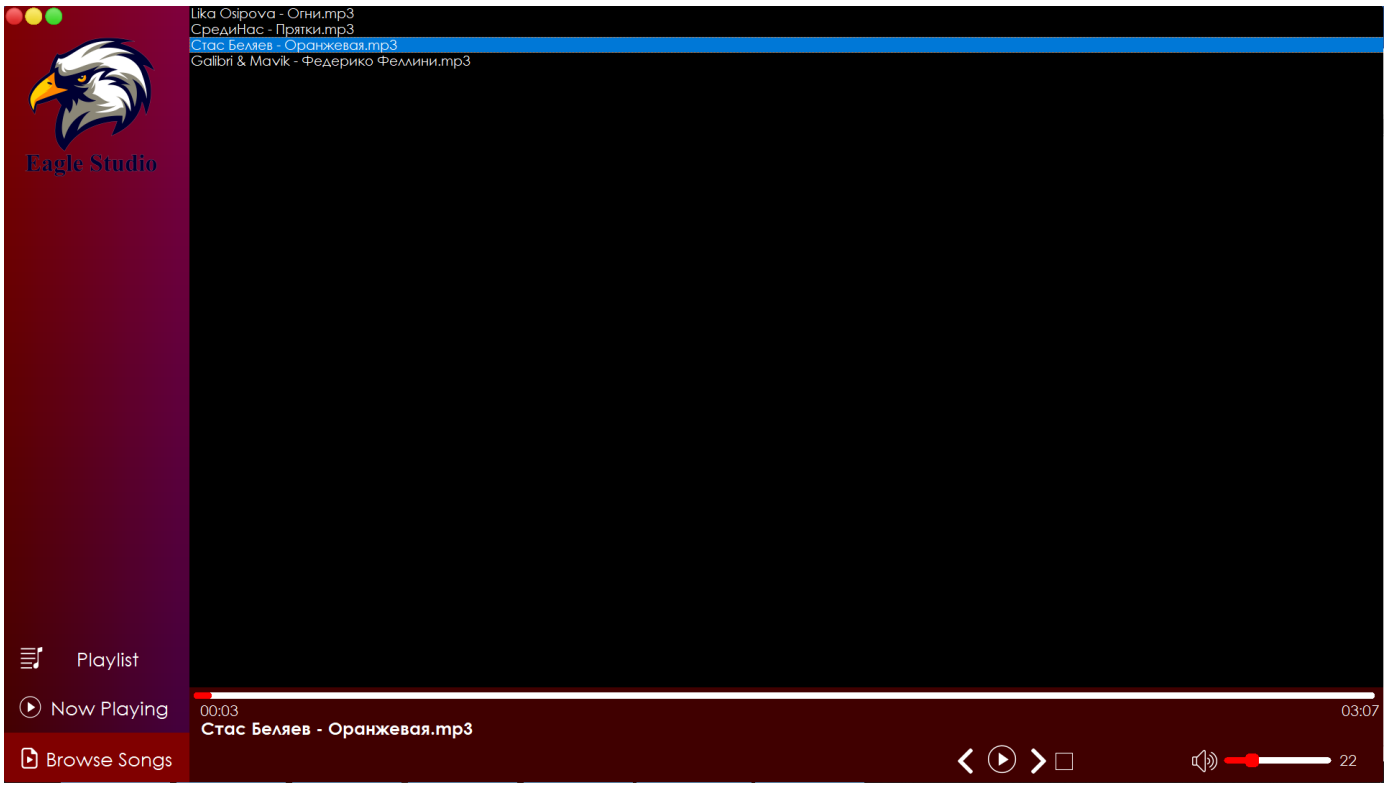

Рис. 2.26. Програма розгорнута у весь екран

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1274;

2. коефіцієнт складності програми – 1,3;

3. коефіцієнт корекції програми в ході її розробки – 0,06;

4. годинна заробітна плата програміста – 60 грн/год;

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;

6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;

7. вартість машино-години ЕОМ – 14 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_д \text{, людино-годин,} \qquad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

$t_д$ – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \text{ де} \tag{3.2}$$

q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q = 1274 \cdot 1{,}2 \cdot (1 + 0{,}06) = 1620{,}53;$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \ldots 85) \cdot K}, \text{ людино-годин,} \tag{3.3}$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

K – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності;

$$t_u = \frac{1620{,}5 \cdot 1{,}2}{85 \cdot 1{,}2} = 19{,}07, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20...25) \cdot K}\ ; \qquad (3.4)$$

$$t_a = \frac{1620{,}53}{20 \cdot 1{,}2} = 67{,}52, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25) \cdot K}; \qquad (3.5)$$

$$t_n = \frac{1620{,}53}{25 \cdot 1{,}2} = 54{,}02, \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4...5) \cdot K}; \qquad (3.6)$$

$$t_n = \frac{1620{,}53}{5 \cdot 1{,}2} = 270{,}08, \text{ людино-годин,}$$

– за умови комплексного налагодження завдання:

$$t^{к}_{отл} = 1{,}2 \cdot t_{отл};$$

(3.7)

$$t^{к}_{отл} = 1{,}2 \cdot 270{,}08 = 324{,}09, \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_д = t_{др} + t_{до};$$

(3.8)

де $t_{др}$ – трудомісткість підготовки матеріалів і рукопису

$$t_д = \frac{Q}{(15...20)\cdot K};$$

(3.9)

$$t_д = \frac{1620,53}{20 \cdot 1,2} = 38,89 \text{ людино-годин.}$$

$t_{до}$– трудомісткість редагування, печатки й оформлення документації

$$t_{до} = 0,75 \cdot t_{др};$$

(3.10)

$$t_{до} = 0,75 \cdot 67,52 = 50,64, \text{ людино-годин.}$$

$$t_д = 67,52 + 50,64 = 118,18, \text{ людино-годин.}$$

Отримаємо трудомісткість розробки програмного забезпечення:

t = 50 + 19,07 + 67,52 + 54,02 + 324,09 + 118,18 = 632,88, людино-годин.

У результаті ми розрахували, що в загальній складності необхідно 632,88 людино-годин для розробки даного програмного забезпечення.

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{ПО} = З_{ЗП} + З_{МВ}\text{, грн,} \qquad (3.11)$$

де $З_{ЗП}$ – заробітна плата виконавців, яка визначається за формулою:

$$З_{ЗП} = t \cdot C_{ПР}\text{, грн,} \qquad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$C_{ПР}$ – середня годинна заробітна плата програміста, грн/година

$$З_{ЗП} = 632{,}88 \cdot 60 = 37972{,}8\text{, грн.}$$

$З_{МВ}$ – Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{МВ} = t_{отл} \cdot C_{М}\text{, грн,} \qquad (3.13)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год.

$C_{МЧ}$ – вартість машино-години ЕОМ, грн/год.

$$З_{МВ} = 324{,}09 \cdot 14 = 4537{,}26 \text{ грн.}$$

$$К_{ПО} = 37972,8 \cdot 4537,26 = 42510,06 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ мес.} \quad (3.14)$$

де B_k- число виконавців;

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні Fp=176 годин).

$$T = \frac{632,88}{1 \cdot 176} = 3,6 \text{ міс.}$$

Висновок: програма розроблена з метою забезпечення користувача зручним і функціональним мультиплеєром. На розробку даного програмного забезпечення піде 632,88 людино-годин. Очікувані витрати на створення програмного забезпечення складатимуть 42510,06 грн.

Тобто, ймовірна очікувана тривалість розробки складатиме 3,1 місяці при стандартному 40-годинному робочому тижні і 176-годинному робочому місяці. Цей термін пов'язаний зі значною кількістю операторів і включає в себе час для дослідження та розробки алгоритму розв'язання задачі, розробку дизайну, створення мультіплеєру та підготовку документації.

ВИСНОВКИ

В даній кваліфікаційній роботі був розроблений медіалеєр основі Bunifu framework.

Це програмне забезпечення призначене для надання можливості зручного прослуховування або перегляду аудіо та відеофайлів багатьох форматів.

Під час виконання даного дипломного проекту були виконані наступні задачі:

– проаналізовано предметну область задачі, що розв’язується;

– проведено порівняння з можливостей існуючих подібних застосунків;

– обрано раціональну структуру і параметри програми;

– написано програмний код додатку;

– розроблено рекомендації щодо використання програми.

Програма працює під керування Windows OS, яка широко використовується цільовою аудиторією продукту. Розробка велась на високорівневій мові програмування C#, яка надає змогу досить точно оперувати ресурсами й дозволила досягти достатніх результатів в швидкості роботи програми. Головною задачею було те, щоб користувач проводив час із задоволенням.

В «Економічному розділі» визначено трудомісткість розробки програмного забезпечення (632,88 чол-год), підраховані витрати на створення програмного забезпечення (42,510 грн.) і гаданий період розробки (3 міс.).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Агуров, Павел C#. Сборник рецептов / Павел Агуров. – М.

2. Албахари, Джозеф C# 3.0. Справочник / Джозеф Албахари , Бен Албахари.

3. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс

4. Бишоп, Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний

5. Вагнер, Билл C# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013

6. Зиборов, В.В. Visual C# 2012 на примерах / В.В. Зиборов

7. Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов.

8. Ишкова, Э. А. Самоучитель C#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013

9. Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами / А.И. Касаткин. - М.: Высшая школа, 2012. - 432 с.

10. Лотка, Рокфорд C# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. - М.: Вильямс, 2010. - 816 с.

11. Мак-Дональд, Мэтью Silverlight 5 с примерами на C# для профессионалов / Мэтью Мак-Дональд. - М.: Вильямс, 2013. - 848 с.

12. Марченко, А. Л. Основы программирования на C# 2.0 / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. - 552 с.

13. Подбельский, В. В. Язык C#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, Инфра-М, 2011. - 384 с.

14. Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс , Майк Гандэрлой. - М.: Век +, Корона-Век, Энтроп, 2010. - 736 с.

15. Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft

.NET Framework 4.0 на языке C# / Джеффри Рихтер, 2013. - 928 с.
16. Смоленцев, Н. К. MATLAB. Программирование на Visual C#, Borland JBuilder, VBA (+ CD-ROM) / Н.К. Смоленцев. - М.: ДМК Пресс, 2011. - 456 с.
17. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 486 с.
18. Троелсен, Эндрю Язык программирования C# 2008 и платформа .NET 3.5 / Эндрю Троелсен. - М.: Вильямс, 2010. - 370 с.
19. Фримен, Адам ASP.NET MVC 3 Framework с примерами на C# для профессионалов / Адам Фримен , Стивен Сандерсон. - М.: Вильямс, 2011. - 672 с.
20. Barrat, James. The latest invention of mankind / James Barrath. - M., 2015. 299 p. (Last visit 14 May 2021)

ДОДАТОК А

КОД ПРОГРАМИ

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MediaPlayer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        string[] paths, files;
        int Startindex = 0;
        string[] FileName, FilePath;
        Boolean playnext = false;

        bool _playing = false;

        public bool isplaying
        {
            get
            {
                return _playing;
```

```
            }
            set
            {
                _playing = value;
                if (_playing)
                {
                    axWindowsMediaPlayer1.Ctlcontrols.pause();  actionbtn.Image  =
play.Image;
                }
                else
                {
                    axWindowsMediaPlayer1.Ctlcontrols.play();  actionbtn.Image  =
pause.Image;
                }
            }
        }
        private void bunifuSlider1_ValueChanged(object sender, EventArgs e)
        {
            axWindowsMediaPlayer1.settings.volume = bunifuSlider1.Value;
            bunifuCustomLabel4.Text = bunifuSlider1.Value.ToString();
        }

        private void btnclose_Click(object sender, EventArgs e)
        {
            Environment.Exit(0);
        }

        private void btnmaximize_Click(object sender, EventArgs e)
        {
            if(this.WindowState==FormWindowState.Maximized)
            {
                this.WindowState = FormWindowState.Normal;
            }
```

```
        else
        {
            this.WindowState = FormWindowState.Maximized;
        }
}

private void btnminimize_Click(object sender, EventArgs e)
{
        this.WindowState = FormWindowState.Minimized;
}

private void bunifuImageButton9_Click(object sender, EventArgs e)
{
        axWindowsMediaPlayer1.Ctlcontrols.play();
}

private void Form1_Load(object sender, EventArgs e)
{
        Startindex = 0;
        playnext = false;
        StopPlayer();
}

private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
        Startindex = listBox1.SelectedIndex;
        playfile(Startindex);
        bunifuCustomLabel3.Text = listBox1.Text;
}

private void btnbrowse_Click(object sender, EventArgs e)
{
        Startindex = 0;
```

```
            playnext = false;
            OpenFileDialog opnFileDlg = new OpenFileDialog();
            opnFileDlg.Multiselect = true;
            opnFileDlg.Filter =
"(mp3,wav,mp4,mov,wmv,mpg,avi,3gp,flv)|*.mp3;*.wav;*.mp4;*.3gp;*.avi;*.mov;*.flv;*.wmv;*.
mpg|all files|*.*";
            if(opnFileDlg.ShowDialog()==DialogResult.OK)
            {
                FileName = opnFileDlg.SafeFileNames;
                FilePath = opnFileDlg.FileNames;
                for(int i=0 ; i <= FileName.Length -1; i++)
                {
                    listBox1.Items.Add(FileName[i]);
                }
                Startindex = 0;
                playfile(0);
            }
        }

        private void btnsong_Click(object sender, EventArgs e)
        {
            listBox1.BringToFront();
        }

        private void btnnowplay_Click(object sender, EventArgs e)
        {
            axWindowsMediaPlayer1.BringToFront();
        }

        private void pause_Click(object sender, EventArgs e)
        {
            axWindowsMediaPlayer1.Ctlcontrols.pause();
        }
```

```
public EventHandler onAction = null;
private void actionbtn_Click(object sender, EventArgs e)
{
    isplaying = !isplaying;
    if (onAction != null)
    {
        onAction.Invoke(this, e);
    }
}

private void bunifuImageButton8_Click(object sender, EventArgs e)
{
    StopPlayer();
}

private void bunifuImageButton5_Click(object sender, EventArgs e)
{
    if(Startindex==listBox1.Items.Count -1)
    {
        Startindex = listBox1.Items.Count - 1;
    }
    else if(Startindex <listBox1.Items.Count)
    {
        Startindex = Startindex + 1;
    }
    playfile(Startindex);
}

private void btnprevious_Click(object sender, EventArgs e)
{
    if(Startindex >0 )
    {
```

```
                Startindex = Startindex - 1;
            }
            playfile(Startindex);
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            bunifuCustomLabel1.Text = axWindowsMediaPlayer1.Ctlcontrols.currentPositionString;
            bunifuCustomLabel2.Text = axWindowsMediaPlayer1.Ctlcontrols.currentItem.durationString.ToString();

if(axWindowsMediaPlayer1.playState==WMPLib.WMPPlayState.wmppsPlaying)
            {
                bunifuProgressBar1.Value = (int)axWindowsMediaPlayer1.Ctlcontrols.currentPosition;
            }
        }

        private void axWindowsMediaPlayer1_PlayStateChange(object sender, AxWMPLib._WMPOCXEvents_PlayStateChangeEvent e)
        {

if(axWindowsMediaPlayer1.playState==WMPLib.WMPPlayState.wmppsPlaying)
            {
                bunifuProgressBar1.MaximumValue = (int)axWindowsMediaPlayer1.Ctlcontrols.currentItem.duration;
                timer1.Start();
            }
            else if (axWindowsMediaPlayer1.playState==WMPLib.WMPPlayState.wmppsPaused)
            {
                timer1.Stop();
```

```
                }
                else if (axWindowsMediaPlayer1.playState == WMPLib.WMPPlayState.wmppsStopped)
                {
                    timer1.Stop();
                    bunifuProgressBar1.Value = 0;
                }
            }

            public void StopPlayer()
            {
                axWindowsMediaPlayer1.Ctlcontrols.stop();
            }

            public void playfile(int playlistindex)
            {
                if(listBox1.Items.Count <=0)
                {
                    return;
                }
                if(playlistindex <0)
                {
                    return;
                }
                axWindowsMediaPlayer1.settings.autoStart = true;
                axWindowsMediaPlayer1.URL = FilePath[playlistindex];
                axWindowsMediaPlayer1.Ctlcontrols.next();
                axWindowsMediaPlayer1.Ctlcontrols.play();
            }
        }
    }
```

Form1.Designer.cs

```
namespace MediaPlayer
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        /// <summary>
        /// Требуемый метод для поддержки конструктора — не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));
```

```
            this.bunifuGradientPanel1 = new Bunifu.Framework.UI.BunifuGradientPanel();
            this.btnsong = new Bunifu.Framework.UI.BunifuFlatButton();
            this.btnnowplay = new Bunifu.Framework.UI.BunifuFlatButton();
            this.btnbrowse = new Bunifu.Framework.UI.BunifuFlatButton();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.btnminimize = new Bunifu.Framework.UI.BunifuImageButton();
            this.btnmaximize = new Bunifu.Framework.UI.BunifuImageButton();
            this.btnclose = new Bunifu.Framework.UI.BunifuImageButton();
            this.panel1 = new System.Windows.Forms.Panel();
            this.btnstop = new Bunifu.Framework.UI.BunifuImageButton();
            this.btnprevious = new Bunifu.Framework.UI.BunifuImageButton();
            this.actionbtn = new Bunifu.Framework.UI.BunifuImageButton();
            this.btnnext = new Bunifu.Framework.UI.BunifuImageButton();
            this.bunifuImageButton4 = new Bunifu.Framework.UI.BunifuImageButton();
            this.bunifuCustomLabel4 = new Bunifu.Framework.UI.BunifuCustomLabel();
            this.bunifuSlider1 = new Bunifu.Framework.UI.BunifuSlider();
            this.bunifuCustomLabel3 = new Bunifu.Framework.UI.BunifuCustomLabel();
            this.bunifuCustomLabel2 = new Bunifu.Framework.UI.BunifuCustomLabel();
            this.bunifuCustomLabel1 = new Bunifu.Framework.UI.BunifuCustomLabel();
            this.bunifuProgressBar1 = new Bunifu.Framework.UI.BunifuProgressBar();
            this.play = new Bunifu.Framework.UI.BunifuImageButton();
            this.pause = new Bunifu.Framework.UI.BunifuImageButton();
            this.bunifuElipse1 = new Bunifu.Framework.UI.BunifuElipse(this.components);
            this.bunifuDragControl1 = new
Bunifu.Framework.UI.BunifuDragControl(this.components);
            this.axWindowsMediaPlayer1 = new AxWMPLib.AxWindowsMediaPlayer();
            this.listBox1 = new System.Windows.Forms.ListBox();
            this.timer1 = new System.Windows.Forms.Timer(this.components);
            this.bunifuGradientPanel1.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnminimize)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnmaximize)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnclose)).BeginInit();
```

```
            this.panel1.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.btnstop)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnprevious)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.actionbtn)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnnext)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.bunifuImageButton4)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.play)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pause)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.axWindowsMediaPlayer1)).BeginInit();
            this.SuspendLayout();
            //
            // bunifuGradientPanel1
            //
            this.bunifuGradientPanel1.BackgroundImage                                   =
((System.Drawing.Image)(resources.GetObject("bunifuGradientPanel1.BackgroundImage")));
            this.bunifuGradientPanel1.BackgroundImageLayout                             =
System.Windows.Forms.ImageLayout.Stretch;
            this.bunifuGradientPanel1.Controls.Add(this.btnsong);
            this.bunifuGradientPanel1.Controls.Add(this.btnnowplay);
            this.bunifuGradientPanel1.Controls.Add(this.btnbrowse);
            this.bunifuGradientPanel1.Controls.Add(this.pictureBox1);
            this.bunifuGradientPanel1.Controls.Add(this.btnminimize);
            this.bunifuGradientPanel1.Controls.Add(this.btnmaximize);
            this.bunifuGradientPanel1.Controls.Add(this.btnclose);
            this.bunifuGradientPanel1.Dock = System.Windows.Forms.DockStyle.Left;
            this.bunifuGradientPanel1.GradientBottomLeft = System.Drawing.Color.Red;
            this.bunifuGradientPanel1.GradientBottomRight = System.Drawing.Color.Maroon;
            this.bunifuGradientPanel1.GradientTopLeft = System.Drawing.Color.Black;
            this.bunifuGradientPanel1.GradientTopRight = System.Drawing.Color.Navy;
            this.bunifuGradientPanel1.Location = new System.Drawing.Point(0, 0);
            this.bunifuGradientPanel1.Name = "bunifuGradientPanel1";
```

```
            this.bunifuGradientPanel1.Quality = 15;
            this.bunifuGradientPanel1.Size = new System.Drawing.Size(258, 726);
            this.bunifuGradientPanel1.TabIndex = 0;
            //
            // btnsong
            //
            this.btnsong.Activecolor = System.Drawing.Color.Maroon;
            this.btnsong.Anchor                                                    =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom    |
System.Windows.Forms.AnchorStyles.Left)));
            this.btnsong.BackColor = System.Drawing.Color.Transparent;
            this.btnsong.BackgroundImageLayout                                     =
System.Windows.Forms.ImageLayout.Stretch;
            this.btnsong.BorderRadius = 0;
            this.btnsong.ButtonText = "Playlist";
            this.btnsong.Cursor = System.Windows.Forms.Cursors.Hand;
            this.btnsong.DisabledColor = System.Drawing.Color.Gray;
            this.btnsong.Iconcolor = System.Drawing.Color.Transparent;
            this.btnsong.Iconimage                                                 =
((System.Drawing.Image)(resources.GetObject("btnsong.Iconimage")));
            this.btnsong.Iconimage_right = null;
            this.btnsong.Iconimage_right_Selected = null;
            this.btnsong.Iconimage_Selected = null;
            this.btnsong.IconMarginLeft = 23;
            this.btnsong.IconMarginRight = 0;
            this.btnsong.IconRightVisible = true;
            this.btnsong.IconRightZoom = 0D;
            this.btnsong.IconVisible = true;
            this.btnsong.IconZoom = 65D;
            this.btnsong.IsTab = true;
            this.btnsong.Location = new System.Drawing.Point(0, 517);
            this.btnsong.Name = "btnsong";
            this.btnsong.Normalcolor = System.Drawing.Color.Transparent;
```

```
            this.btnsong.OnHovercolor =
System.Drawing.Color.FromArgb(((int)(((byte)(192)))), ((int)(((byte)(0)))), ((int)(((byte)(0)))));
            this.btnsong.OnHoverTextColor = System.Drawing.Color.White;
            this.btnsong.selected = false;
            this.btnsong.Size = new System.Drawing.Size(258, 70);
            this.btnsong.TabIndex = 6;
            this.btnsong.Text = "Playlist";
            this.btnsong.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
            this.btnsong.Textcolor = System.Drawing.Color.White;
            this.btnsong.TextFont = new System.Drawing.Font("Century Gothic", 20.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.btnsong.Click += new System.EventHandler(this.btnsong_Click);
            //
            // btnnowplay
            //
            this.btnnowplay.Activecolor = System.Drawing.Color.Maroon;
            this.btnnowplay.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)));
            this.btnnowplay.BackColor = System.Drawing.Color.Transparent;
            this.btnnowplay.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
            this.btnnowplay.BorderRadius = 0;
            this.btnnowplay.ButtonText = "Now Playing";
            this.btnnowplay.Cursor = System.Windows.Forms.Cursors.Hand;
            this.btnnowplay.DisabledColor = System.Drawing.Color.Gray;
            this.btnnowplay.Iconcolor = System.Drawing.Color.Transparent;
            this.btnnowplay.Iconimage =
((System.Drawing.Image)(resources.GetObject("btnnowplay.Iconimage")));
            this.btnnowplay.Iconimage_right = null;
            this.btnnowplay.Iconimage_right_Selected = null;
            this.btnnowplay.Iconimage_Selected = null;
            this.btnnowplay.IconMarginLeft = 0;
```

```
            this.btnnowplay.IconMarginRight = 0;
            this.btnnowplay.IconRightVisible = true;
            this.btnnowplay.IconRightZoom = 0D;
            this.btnnowplay.IconVisible = true;
            this.btnnowplay.IconZoom = 65D;
            this.btnnowplay.IsTab = true;
            this.btnnowplay.Location = new System.Drawing.Point(0, 584);
            this.btnnowplay.Name = "btnnowplay";
            this.btnnowplay.Normalcolor = System.Drawing.Color.Transparent;
            this.btnnowplay.OnHovercolor =
System.Drawing.Color.FromArgb(((int)(((byte)(192)))), ((int)(((byte)(0)))), ((int)(((byte)(0)))));
            this.btnnowplay.OnHoverTextColor = System.Drawing.Color.White;
            this.btnnowplay.selected = false;
            this.btnnowplay.Size = new System.Drawing.Size(267, 72);
            this.btnnowplay.TabIndex = 5;
            this.btnnowplay.Text = "Now Playing";
            this.btnnowplay.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
            this.btnnowplay.Textcolor = System.Drawing.Color.White;
            this.btnnowplay.TextFont = new System.Drawing.Font("Century Gothic", 20.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.btnnowplay.Click += new System.EventHandler(this.btnnowplay_Click);
            //
            // btnbrowse
            //
            this.btnbrowse.Activecolor = System.Drawing.Color.Maroon;
            this.btnbrowse.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)));
            this.btnbrowse.BackColor = System.Drawing.Color.Transparent;
            this.btnbrowse.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
            this.btnbrowse.BorderRadius = 0;
            this.btnbrowse.ButtonText = "Browse Songs";
```

```
            this.btnbrowse.Cursor = System.Windows.Forms.Cursors.Hand;
            this.btnbrowse.DisabledColor = System.Drawing.Color.Gray;
            this.btnbrowse.Iconcolor = System.Drawing.Color.Transparent;
            this.btnbrowse.Iconimage =
((System.Drawing.Image)(resources.GetObject("btnbrowse.Iconimage")));
            this.btnbrowse.Iconimage_right = null;
            this.btnbrowse.Iconimage_right_Selected = null;
            this.btnbrowse.Iconimage_Selected = null;
            this.btnbrowse.IconMarginLeft = 0;
            this.btnbrowse.IconMarginRight = 0;
            this.btnbrowse.IconRightVisible = true;
            this.btnbrowse.IconRightZoom = 0D;
            this.btnbrowse.IconVisible = true;
            this.btnbrowse.IconZoom = 60D;
            this.btnbrowse.IsTab = true;
            this.btnbrowse.Location = new System.Drawing.Point(0, 656);
            this.btnbrowse.Name = "btnbrowse";
            this.btnbrowse.Normalcolor = System.Drawing.Color.Transparent;
            this.btnbrowse.OnHovercolor =
System.Drawing.Color.FromArgb(((int)(((byte)(192)))), ((int)(((byte)(0)))), ((int)(((byte)(0)))));
            this.btnbrowse.OnHoverTextColor = System.Drawing.Color.White;
            this.btnbrowse.selected = false;
            this.btnbrowse.Size = new System.Drawing.Size(267, 70);
            this.btnbrowse.TabIndex = 4;
            this.btnbrowse.Text = "Browse Songs";
            this.btnbrowse.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
            this.btnbrowse.Textcolor = System.Drawing.Color.White;
            this.btnbrowse.TextFont = new System.Drawing.Font("Century Gothic", 20.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.btnbrowse.Click += new System.EventHandler(this.btnbrowse_Click);
            //
            // pictureBox1
            //
```

```
            this.pictureBox1.BackColor = System.Drawing.Color.Transparent;
            this.pictureBox1.Image =
((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));
            this.pictureBox1.Location = new System.Drawing.Point(12, 29);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(220, 214);
            this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.pictureBox1.TabIndex = 3;
            this.pictureBox1.TabStop = false;
            //
            // btnminimize
            //
            this.btnminimize.BackColor = System.Drawing.Color.Transparent;
            this.btnminimize.Cursor = System.Windows.Forms.Cursors.Hand;
            this.btnminimize.Image =
((System.Drawing.Image)(resources.GetObject("btnminimize.Image")));
            this.btnminimize.ImageActive = null;
            this.btnminimize.Location = new System.Drawing.Point(57, 2);
            this.btnminimize.Name = "btnminimize";
            this.btnminimize.Size = new System.Drawing.Size(25, 25);
            this.btnminimize.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.btnminimize.TabIndex = 2;
            this.btnminimize.TabStop = false;
            this.btnminimize.Zoom = 10;
            this.btnminimize.Click += new System.EventHandler(this.btnminimize_Click);
            //
            // btnmaximize
            //
            this.btnmaximize.BackColor = System.Drawing.Color.Transparent;
            this.btnmaximize.Cursor = System.Windows.Forms.Cursors.Hand;
            this.btnmaximize.Image =
((System.Drawing.Image)(resources.GetObject("btnmaximize.Image")));
            this.btnmaximize.ImageActive = null;
```

```
            this.btnmaximize.Location = new System.Drawing.Point(29, 2);
            this.btnmaximize.Name = "btnmaximize";
            this.btnmaximize.Size = new System.Drawing.Size(25, 25);
            this.btnmaximize.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.btnmaximize.TabIndex = 1;
            this.btnmaximize.TabStop = false;
            this.btnmaximize.Zoom = 10;
            this.btnmaximize.Click += new System.EventHandler(this.btnmaximize_Click);
            //
            // btnclose
            //
            this.btnclose.BackColor = System.Drawing.Color.Transparent;
            this.btnclose.Cursor = System.Windows.Forms.Cursors.Hand;
            this.btnclose.Image                                                    =
((System.Drawing.Image)(resources.GetObject("btnclose.Image")));
            this.btnclose.ImageActive = null;
            this.btnclose.Location = new System.Drawing.Point(2, 2);
            this.btnclose.Name = "btnclose";
            this.btnclose.Size = new System.Drawing.Size(25, 25);
            this.btnclose.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.btnclose.TabIndex = 0;
            this.btnclose.TabStop = false;
            this.btnclose.Zoom = 10;
            this.btnclose.Click += new System.EventHandler(this.btnclose_Click);
            //
            // panel1
            //
            this.panel1.BackColor    =    System.Drawing.Color.FromArgb(((int)(((byte)(64)))),
((int)(((byte)(0)))), ((int)(((byte)(0)))));
            this.panel1.Controls.Add(this.btnstop);
            this.panel1.Controls.Add(this.btnprevious);
            this.panel1.Controls.Add(this.actionbtn);
            this.panel1.Controls.Add(this.btnnext);
```

```
            this.panel1.Controls.Add(this.bunifuImageButton4);
            this.panel1.Controls.Add(this.bunifuCustomLabel4);
            this.panel1.Controls.Add(this.bunifuSlider1);
            this.panel1.Controls.Add(this.bunifuCustomLabel3);
            this.panel1.Controls.Add(this.bunifuCustomLabel2);
            this.panel1.Controls.Add(this.bunifuCustomLabel1);
            this.panel1.Controls.Add(this.bunifuProgressBar1);
            this.panel1.Controls.Add(this.play);
            this.panel1.Controls.Add(this.pause);
            this.panel1.Dock = System.Windows.Forms.DockStyle.Bottom;
            this.panel1.Location = new System.Drawing.Point(258, 593);
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(1014, 133);
            this.panel1.TabIndex = 1;
            //
            // btnstop
            //
            this.btnstop.Anchor                                                    =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top          |
System.Windows.Forms.AnchorStyles.Right)));
            this.btnstop.BackColor = System.Drawing.Color.Transparent;
            this.btnstop.Image                                                     =
((System.Drawing.Image)(resources.GetObject("btnstop.Image")));
            this.btnstop.ImageActive = null;
            this.btnstop.Location = new System.Drawing.Point(557, 91);
            this.btnstop.Name = "btnstop";
            this.btnstop.Size = new System.Drawing.Size(25, 25);
            this.btnstop.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.btnstop.TabIndex = 11;
            this.btnstop.TabStop = false;
            this.btnstop.Zoom = 10;
            this.btnstop.Click += new System.EventHandler(this.bunifuImageButton8_Click);
            //
```

```
            // btnprevious
            //
            this.btnprevious.Anchor                                                    =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top                |
System.Windows.Forms.AnchorStyles.Right)));
            this.btnprevious.BackColor = System.Drawing.Color.Transparent;
            this.btnprevious.Image                                                     =
((System.Drawing.Image)(resources.GetObject("btnprevious.Image")));
            this.btnprevious.ImageActive = null;
            this.btnprevious.Location = new System.Drawing.Point(414, 85);
            this.btnprevious.Name = "btnprevious";
            this.btnprevious.Size = new System.Drawing.Size(35, 35);
            this.btnprevious.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.btnprevious.TabIndex = 10;
            this.btnprevious.TabStop = false;
            this.btnprevious.Zoom = 10;
            this.btnprevious.Click += new System.EventHandler(this.btnprevious_Click);
            //
            // actionbtn
            //
            this.actionbtn.Anchor                                                      =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top                |
System.Windows.Forms.AnchorStyles.Right)));
            this.actionbtn.BackColor = System.Drawing.Color.Transparent;
            this.actionbtn.Image                                                       =
((System.Drawing.Image)(resources.GetObject("actionbtn.Image")));
            this.actionbtn.ImageActive = null;
            this.actionbtn.Location = new System.Drawing.Point(462, 79);
            this.actionbtn.Name = "actionbtn";
            this.actionbtn.Size = new System.Drawing.Size(41, 43);
            this.actionbtn.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.actionbtn.TabIndex = 9;
            this.actionbtn.TabStop = false;
```

```
            this.actionbtn.Zoom = 10;
            this.actionbtn.Click += new System.EventHandler(this.actionbtn_Click);
            //
            // btnnext
            //
            this.btnnext.Anchor                                                    =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top    |
System.Windows.Forms.AnchorStyles.Right)));
            this.btnnext.BackColor = System.Drawing.Color.Transparent;
            this.btnnext.Image                                                     =
((System.Drawing.Image)(resources.GetObject("btnnext.Image")));
            this.btnnext.ImageActive = null;
            this.btnnext.Location = new System.Drawing.Point(516, 85);
            this.btnnext.Name = "btnnext";
            this.btnnext.Size = new System.Drawing.Size(35, 35);
            this.btnnext.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.btnnext.TabIndex = 8;
            this.btnnext.TabStop = false;
            this.btnnext.Zoom = 10;
            this.btnnext.Click += new System.EventHandler(this.bunifuImageButton5_Click);
            //
            // bunifuImageButton4
            //
            this.bunifuImageButton4.Anchor                                         =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top    |
System.Windows.Forms.AnchorStyles.Right)));
            this.bunifuImageButton4.BackColor = System.Drawing.Color.Transparent;
            this.bunifuImageButton4.Image                                          =
((System.Drawing.Image)(resources.GetObject("bunifuImageButton4.Image")));
            this.bunifuImageButton4.ImageActive = null;
            this.bunifuImageButton4.Location = new System.Drawing.Point(747, 85);
            this.bunifuImageButton4.Name = "bunifuImageButton4";
            this.bunifuImageButton4.Size = new System.Drawing.Size(35, 35);
```

```
            this.bunifuImageButton4.SizeMode                                    =
System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.bunifuImageButton4.TabIndex = 7;
            this.bunifuImageButton4.TabStop = false;
            this.bunifuImageButton4.Zoom = 10;
            //
            // bunifuCustomLabel4
            //
            this.bunifuCustomLabel4.Anchor                                      =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top     |
System.Windows.Forms.AnchorStyles.Right)));
            this.bunifuCustomLabel4.AutoSize = true;
            this.bunifuCustomLabel4.Font = new System.Drawing.Font("Century Gothic",
15.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.bunifuCustomLabel4.ForeColor = System.Drawing.Color.White;
            this.bunifuCustomLabel4.Location = new System.Drawing.Point(948, 89);
            this.bunifuCustomLabel4.Name = "bunifuCustomLabel4";
            this.bunifuCustomLabel4.Size = new System.Drawing.Size(46, 24);
            this.bunifuCustomLabel4.TabIndex = 4;
            this.bunifuCustomLabel4.Text = "100";
            //
            // bunifuSlider1
            //
            this.bunifuSlider1.Anchor                                           =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top     |
System.Windows.Forms.AnchorStyles.Right)));
            this.bunifuSlider1.BackColor = System.Drawing.Color.Transparent;
            this.bunifuSlider1.BackgroudColor = System.Drawing.Color.White;
            this.bunifuSlider1.BorderRadius = 10;
            this.bunifuSlider1.IndicatorColor = System.Drawing.Color.Red;
            this.bunifuSlider1.Location = new System.Drawing.Point(791, 89);
            this.bunifuSlider1.MaximumValue = 100;
            this.bunifuSlider1.Name = "bunifuSlider1";
```

```
            this.bunifuSlider1.Size = new System.Drawing.Size(151, 30);
            this.bunifuSlider1.TabIndex = 2;
            this.bunifuSlider1.Value = 100;
            this.bunifuSlider1.ValueChanged += new
System.EventHandler(this.bunifuSlider1_ValueChanged);
            //
            // bunifuCustomLabel3
            //
            this.bunifuCustomLabel3.AutoSize = true;
            this.bunifuCustomLabel3.Font = new System.Drawing.Font("Century Gothic", 18F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.bunifuCustomLabel3.ForeColor = System.Drawing.Color.White;
            this.bunifuCustomLabel3.Location = new System.Drawing.Point(11, 43);
            this.bunifuCustomLabel3.Name = "bunifuCustomLabel3";
            this.bunifuCustomLabel3.Size = new System.Drawing.Size(148, 28);
            this.bunifuCustomLabel3.TabIndex = 3;
            this.bunifuCustomLabel3.Text = "Song Name";
            //
            // bunifuCustomLabel2
            //
            this.bunifuCustomLabel2.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Right)));
            this.bunifuCustomLabel2.AutoSize = true;
            this.bunifuCustomLabel2.Font = new System.Drawing.Font("Century Gothic",
15.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.bunifuCustomLabel2.ForeColor = System.Drawing.Color.White;
            this.bunifuCustomLabel2.Location = new System.Drawing.Point(951, 20);
            this.bunifuCustomLabel2.Name = "bunifuCustomLabel2";
            this.bunifuCustomLabel2.Size = new System.Drawing.Size(51, 24);
            this.bunifuCustomLabel2.TabIndex = 2;
            this.bunifuCustomLabel2.Text = "0.00";
            //
```

```
            // bunifuCustomLabel1
            //
            this.bunifuCustomLabel1.AutoSize = true;
            this.bunifuCustomLabel1.Font = new System.Drawing.Font("Century Gothic",
15.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.bunifuCustomLabel1.ForeColor = System.Drawing.Color.White;
            this.bunifuCustomLabel1.Location = new System.Drawing.Point(11, 20);
            this.bunifuCustomLabel1.Name = "bunifuCustomLabel1";
            this.bunifuCustomLabel1.Size = new System.Drawing.Size(51, 24);
            this.bunifuCustomLabel1.TabIndex = 1;
            this.bunifuCustomLabel1.Text = "0.00";
            //
            // bunifuProgressBar1
            //
            this.bunifuProgressBar1.Anchor                                          =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top           |
System.Windows.Forms.AnchorStyles.Left)
            | System.Windows.Forms.AnchorStyles.Right)));
            this.bunifuProgressBar1.BackColor = System.Drawing.Color.White;
            this.bunifuProgressBar1.BorderRadius = 5;
            this.bunifuProgressBar1.Location = new System.Drawing.Point(6, 7);
            this.bunifuProgressBar1.MaximumValue = 100;
            this.bunifuProgressBar1.Name = "bunifuProgressBar1";
            this.bunifuProgressBar1.ProgressColor = System.Drawing.Color.Red;
            this.bunifuProgressBar1.Size = new System.Drawing.Size(996, 10);
            this.bunifuProgressBar1.TabIndex = 0;
            this.bunifuProgressBar1.Value = 0;
            //
            // play
            //
            this.play.Anchor                                                        =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top             |
System.Windows.Forms.AnchorStyles.Right)));
```

```
            this.play.BackColor = System.Drawing.Color.Transparent;
            this.play.Image = ((System.Drawing.Image)(resources.GetObject("play.Image")));
            this.play.ImageActive = null;
            this.play.Location = new System.Drawing.Point(367, 92);
            this.play.Name = "play";
            this.play.Size = new System.Drawing.Size(122, 21);
            this.play.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.play.TabIndex = 13;
            this.play.TabStop = false;
            this.play.Zoom = 10;
            this.play.Click += new System.EventHandler(this.bunifuImageButton9_Click);
            //
            // pause
            //
            this.pause.Anchor                                                    =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top                |
System.Windows.Forms.AnchorStyles.Right)));
            this.pause.BackColor = System.Drawing.Color.Transparent;
            this.pause.Image = ((System.Drawing.Image)(resources.GetObject("pause.Image")));
            this.pause.ImageActive = null;
            this.pause.Location = new System.Drawing.Point(471, 92);
            this.pause.Name = "pause";
            this.pause.Size = new System.Drawing.Size(122, 21);
            this.pause.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
            this.pause.TabIndex = 12;
            this.pause.TabStop = false;
            this.pause.Zoom = 10;
            this.pause.Click += new System.EventHandler(this.pause_Click);
            //
            // bunifuElipse1
            //
            this.bunifuElipse1.ElipseRadius = 5;
            this.bunifuElipse1.TargetControl = this;
```

```
            //
            // bunifuDragControl1
            //
            this.bunifuDragControl1.Fixed = true;
            this.bunifuDragControl1.Horizontal = true;
            this.bunifuDragControl1.TargetControl = this.bunifuGradientPanel1;
            this.bunifuDragControl1.Vertical = true;
            //
            // axWindowsMediaPlayer1
            //
            this.axWindowsMediaPlayer1.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
            | System.Windows.Forms.AnchorStyles.Left)
            | System.Windows.Forms.AnchorStyles.Right)));
            this.axWindowsMediaPlayer1.Enabled = true;
            this.axWindowsMediaPlayer1.Location = new System.Drawing.Point(258, 2);
            this.axWindowsMediaPlayer1.Name = "axWindowsMediaPlayer1";
            this.axWindowsMediaPlayer1.OcxState =
((System.Windows.Forms.AxHost.State)(resources.GetObject("axWindowsMediaPlayer1.OcxState
")));
            this.axWindowsMediaPlayer1.Size = new System.Drawing.Size(1014, 591);
            this.axWindowsMediaPlayer1.TabIndex = 2;
            this.axWindowsMediaPlayer1.PlayStateChange += new
AxWMPLib._WMPOCXEvents_PlayStateChangeEventHandler(this.axWindowsMediaPlayer1_Pla
yStateChange);
            //
            // listBox1
            //
            this.listBox1.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
            | System.Windows.Forms.AnchorStyles.Left)
```

```
            | System.Windows.Forms.AnchorStyles.Right)));
            this.listBox1.BackColor = System.Drawing.Color.Black;
            this.listBox1.BorderStyle = System.Windows.Forms.BorderStyle.None;
            this.listBox1.Font = new System.Drawing.Font("Century Gothic", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.listBox1.ForeColor = System.Drawing.Color.White;
            this.listBox1.FormattingEnabled = true;
            this.listBox1.ItemHeight = 22;
            this.listBox1.Location = new System.Drawing.Point(258, -1);
            this.listBox1.Name = "listBox1";
            this.listBox1.Size = new System.Drawing.Size(1014, 594);
            this.listBox1.TabIndex = 3;
            this.listBox1.SelectedIndexChanged += new
System.EventHandler(this.listBox1_SelectedIndexChanged);
            //
            // timer1
            //
            this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(1272, 726);
            this.Controls.Add(this.listBox1);
            this.Controls.Add(this.axWindowsMediaPlayer1);
            this.Controls.Add(this.panel1);
            this.Controls.Add(this.bunifuGradientPanel1);
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
            this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.Name = "Form1";
            this.Text = "Eagle Studio";
            this.Load += new System.EventHandler(this.Form1_Load);
```

```
            this.bunifuGradientPanel1.ResumeLayout(false);
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnminimize)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnmaximize)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnclose)).EndInit();
            this.panel1.ResumeLayout(false);
            this.panel1.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.btnstop)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnprevious)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.actionbtn)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.btnnext)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.bunifuImageButton4)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.play)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pause)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.axWindowsMediaPlayer1)).EndInit();
            this.ResumeLayout(false);

        }

        #endregion

        private Bunifu.Framework.UI.BunifuGradientPanel bunifuGradientPanel1;
        private Bunifu.Framework.UI.BunifuImageButton btnclose;
        private System.Windows.Forms.Panel panel1;
        private Bunifu.Framework.UI.BunifuImageButton btnminimize;
        private Bunifu.Framework.UI.BunifuImageButton btnmaximize;
        private Bunifu.Framework.UI.BunifuFlatButton btnsong;
        private Bunifu.Framework.UI.BunifuFlatButton btnnowplay;
        private Bunifu.Framework.UI.BunifuFlatButton btnbrowse;
        private System.Windows.Forms.PictureBox pictureBox1;
        private Bunifu.Framework.UI.BunifuElipse bunifuElipse1;
        private Bunifu.Framework.UI.BunifuDragControl bunifuDragControl1;
```

```
        private Bunifu.Framework.UI.BunifuCustomLabel bunifuCustomLabel2;
        private Bunifu.Framework.UI.BunifuCustomLabel bunifuCustomLabel1;
        private Bunifu.Framework.UI.BunifuProgressBar bunifuProgressBar1;
        private Bunifu.Framework.UI.BunifuCustomLabel bunifuCustomLabel4;
        private Bunifu.Framework.UI.BunifuSlider bunifuSlider1;
        private Bunifu.Framework.UI.BunifuCustomLabel bunifuCustomLabel3;
        private Bunifu.Framework.UI.BunifuImageButton bunifuImageButton4;
        private Bunifu.Framework.UI.BunifuImageButton btnprevious;
        private Bunifu.Framework.UI.BunifuImageButton actionbtn;
        private Bunifu.Framework.UI.BunifuImageButton btnnext;
        private Bunifu.Framework.UI.BunifuImageButton btnstop;
        private Bunifu.Framework.UI.BunifuImageButton play;
        private Bunifu.Framework.UI.BunifuImageButton pause;
        private AxWMPLib.AxWindowsMediaPlayer axWindowsMediaPlayer1;
        private System.Windows.Forms.ListBox listBox1;
        private System.Windows.Forms.Timer timer1;
    }
}
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MediaPlayer
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
```

```
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Settings.Designer.cs

```
namespace MediaPlayer.Properties
{
    [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator", "11.0.0.0")]
    internal sealed partial class Settings : global::System.Configuration.ApplicationSettingsBase
    {

        private static Settings defaultInstance = ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new Settings())));

        public static Settings Default
        {
            get
            {
                return defaultInstance;
            }
        }
    }
}
```

Resources.Designer.cs

```
namespace MediaPlayer.Properties
{
```

```
    /// <summary>
    ///     Класс ресурсов со строгим типом для поиска локализованных строк и пр.
    /// </summary>
    // класс с помощью таких средств, как ResGen или Visual Studio.
    // Для добавления или удаления члена измените файл .ResX, а затем перезапустите ResGen
    // с параметром /str или заново постройте свой VS-проект.

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder", "4.0.0.0")]
    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
    internal class Resources
    {

        private static global::System.Resources.ResourceManager resourceMan;

        private static global::System.Globalization.CultureInfo resourceCulture;


[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance", "CA1811:AvoidUncalledPrivateCode")]
        internal Resources()
        {
        }

        /// <summary>
        ///     Возврат кэшированного экземпляра ResourceManager, используемого этим классом.
        /// </summary>

[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
```

```
        internal static global::System.Resources.ResourceManager ResourceManager
        {
            get
            {
                if ((resourceMan == null))
                {
                    global::System.Resources.ResourceManager temp = new
global::System.Resources.ResourceManager("MediaPlayer.Properties.Resources",
typeof(Resources).Assembly);
                    resourceMan = temp;
                }
                return resourceMan;
            }
        }

        /// <summary>
        ///   Переопределяет свойство CurrentUICulture текущего потока для всех
        ///   подстановки ресурсов с помощью этого класса ресурсов со строгим типом.
        /// </summary>

[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.Edi
torBrowsableState.Advanced)]
        internal static global::System.Globalization.CultureInfo Culture
        {
            get
            {
                return resourceCulture;
            }
            set
            {
                resourceCulture = value;
            }
        }
```

```
        }
}
```

AssemblyInfo.cs

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// Общие сведения об этой сборке предоставляются следующим набором
// набора атрибутов. Измените значения этих атрибутов для изменения сведений,
// связанных со сборкой.
[assembly: AssemblyTitle("MediaPlayer")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("MediaPlayer")]
[assembly: AssemblyCopyright("Copyright ©  2021")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Установка значения False для параметра ComVisible делает типы в этой сборке невидимыми
// для компонентов COM. Если необходимо обратиться к типу в этой сборке через
// COM, следует установить атрибут ComVisible в TRUE для этого типа.
[assembly: ComVisible(false)]

// Следующий GUID служит для идентификации библиотеки типов, если этот проект будет видимым для COM
[assembly: Guid("2026bbaf-68f4-4854-878a-67d797244dbb")]
```

ДОДАТОК Б

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ДОДАТОК В

ПРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім’я файлу	Опис
Пояснювальні документи	
Диплом_Сєров.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word
Диплом_Сєров.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Програма.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Сєров.ppt	Презентація кваліфікаційної роботи

ВІДГУК

на кваліфікаційну роботу бакалавра

на тему:

"Розробка Media Player на C# з використанням Bunifu UI framework ".
студента групи 122-18СК-2 Сєрова Даниїла Олексійовича

Розроблене в кваліфікаційній роботі програмне забезпечення призначене для зручного прослуховування або перегляду аудіо та відеофайлів багатьох форматів, яке спроможне організовувати користувача з додатком, яке використовуватиме самий користувач.

Актуальність розробленого програмного продукту полягає в створенні такого додатку, який матиме попит у всій аудиторії.

Для реалізації медіалеєру була використана мова програмування C#. В якості фреймворку для інтерфейсу було обрано технологію Bunifu UI. Для програми візуалізатора використовується WinForms.

Для вирішення поставлених задач при розробці додатку були здійснені наступні дії:

1. Розробка логічної моделі програми.
2. Проектування додатку.
3. Розробка архітектури програмного забезпечення.
4. Розробка простого і зрозумілого інтерфейсу програми.

Додаток розроблений в середовищі Microsoft Visual Studio 2019.

Працездатність представленої програми підтверджена налагоджувальними випробуваннями та тестуванням програми.

Практична значимість створення даного програмного продукту полягає в забезпеченні користувача якісним і зручним мультиплеєром.

В економічному розділі визначено трудомісткість розробленого додатку, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра за напрямом підготовки 122 Комп’ютерні науки.

Оформлення пояснювальної записки до кваліфікаційної роботи виконано відповідно до стандартів на програмну документацію. Недоліком розглянутої роботи є неповний опис алгоритму функціонування системи.

Кваліфікаційна робота виконана самостійно та заслуговує оцінки «добре», а студенту Сєрову Даниїлу Олексійовичу присвоєння кваліфікації бакалавра за спеціальністю «фахівець з розробки та тестування програмного забезпечення».

Керівник кваліфікаційної роботи
ас. каф. ПЗКС, к.т.н. **Приходченко С.Д.**

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

на тему:

" Розробка Media Player на C# з використанням Bunifu UI framework."

студента групи 122-18СК-2 Сєрова Даниїла Олексійовича

Кваліфікаційна робота на тему «Розробка Media Player на C# з використанням Bunifu UI framework» виконана в повному обсязі в співвідношенні з технічним завданням.

Основною метою програми є організація користування, що базується на взаємодії користувача і програми за допомогою візуального інтерфейсу. Було розроблено медіалеєр з інтерфейсом Bunifu UI.

В якості інструмента для реалізації була використана мова програмування C#. Сам проект був реалізований за допомогою framework Bunifu.

У вступі проведений аналіз аналогів вирішення проблеми в цілому, аналоги конкретного методу вирішення проблеми, перераховані їх переваги й недоліки. На основі них були сформовані вимоги до програного забезпечення. В теоретичній частині наявні дані про мету розробки, сферу застосування продукту та його актуальність. Описані функціональне призначення й логічна структура. Вибір технологій й підхід до реалізації повністю обґрунтовані.

Результати реалізації проекту добре описані й підтверджені тестуванням. Проектна документація описує усі можливі варіанти взаємодії з продуктом.

Список літератури налічує більше 20 джерел, що свідчить про вміння працювати з літературою та іншими джерелами інформації.

Робота оцінюється на 82 бала «добре».

Рецензент кваліфікаційної роботи

к.т.н., доцент каф. БІТ **Герасіна О.В.**