**Реферат**

Пояснительная записка к курсовой работе содержит: 49 c., 34 рис., 3 источника.

VISUAL STUDIO 2022, C#, ПРОГРАММЫ, GIF-анимации, ФАЙЛЫ.

Цель работы состоит в разработке интерактивного приложения, предназначенного для создания, настройки и сохранения GIF-анимации из набора изображений, в целях развития навыков прикладного и профессионального программирования.

**Содержание**

[**Введение** 4](#_Toc150951663)

[**1 Нормативные ссылки** 5](#_Toc150951664)

[**2 Описание предметной области** 6](#_Toc150951665)

[**2.1 Общие сведения** 6](#_Toc150951666)

[**2.2 Сведения из теории** 7](#_Toc150951667)

[**2.3 Библиотеки, реализующие функцию создания GIF-анимации** 8](#_Toc150951668)

[**2.4 Постановка задачи** 11](#_Toc150951669)

[**3 Технология разработки программы** 12](#_Toc150951670)

[**3.1 Описание алгоритмов решения** 12](#_Toc150951671)

[**3.2 Макет программы** 12](#_Toc150951672)

[**3.3 Описание программы** 14](#_Toc150951673)

[**3.4 Результат выполнения программы** 14](#_Toc150951674)

[**Заключение** 27](#_Toc150951675)

[**Список использованных источников** 28](#_Toc150951676)

[**Приложение А. Тест на антиплагиат** 29](#_Toc150951677)

# **Введение**

Данный документ подготовлен как пояснительная записка к курсовой работе. Тема курсовой работы ‒ «Создание GIF-анимаций из набора изображений». Выполнение данной курсовой работы позволяет студенту второго курса углубить свои знания связанные с C# и средой разработки .NET, а также улучшить навыки проектирования и разработки пользовательских приложений. Поэтому эта тема является актуальной.

Цель работы заключается в разработке интерактивного пользовательского приложений, предназначенного для создания GIF-анимаций из набора изображений.

Первый раздел содержит ссылки на использованные при составлении пояснительной записки ГОСТы.

Каждый из разделов со второго по третьей пояснительной записки к курсовой работе содержит основные сведения по теме курсовой работы. Во втором разделе представлена теоретическая информация, связанная с таким понятием, как GIF-анимация и идеи по реализации программного продукта. В третьем разделе описана технология разработки и представлено техническое описание программного продукта.

Задачи:

1. Рассмотреть теоретическую часть GIF-анимация
2. Выбрать расширение для создания GIF-анимация
3. Разработать программу по созданию GIF-анимация
4. Оформить интерфейс программы для интерактивного использования

При решении задач курсовой работы были использованы следующие методы: анализ и синтез, моделирование, алгоритмизация.

**1 Нормативные ссылки**

В данной пояснительной записке использованы ссылки на следующие стандарты:

–ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам;

–ГОСТ Р.7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления;

–ГОСТ 7.12-93 СИБИД. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила.

# **2 Описание предметной области**

## **2.1 Общие сведения**

Гиф-анимация представляет собой формат изображений, который позволяет создавать анимированные картинки путем последовательного отображения нескольких кадров. Формат GIF (Graphics Interchange Format) был разработан компанией CompuServe в конце 1980-х годов и быстро стал популярным благодаря своей простоте и универсальности.

Особенности формата GIF:

1. Анимация: Одним из ключевых элементов формата GIF является способность хранить несколько изображений в одном файле и проигрывать их последовательно, создавая впечатление анимации.
2. Индексированные цвета: GIF поддерживает индексированные цвета, что позволяет использовать до 256 цветов из общего палитры 24-битного RGB цвета. Это делает формат компактным и удобным для передачи через интернет.
3. Прозрачность: GIF поддерживает прозрачность, что позволяет одному цвету из палитры считаться прозрачным. Это полезно, например, для создания изображений с прозрачными фонами.
4. Циклическое воспроизведение: GIF-анимации могут быть настроены на циклическое воспроизведение, что означает, что анимация будет повторяться после завершения последнего кадра.

Применение GIF-анимации:

1. Социальные сети: GIF-анимации широко используются для создания коротких, забавных анимаций, которые легко распространяются через социальные сети.
2. Веб-дизайн: Анимированные GIF-изображения активно применяются в веб-дизайне для создания привлекательных и интерактивных элементов на веб-сайтах.
3. Эмодзи и стикеры: Формат GIF часто используется для создания анимированных эмодзи и стикеров, добавляя выразительность в текстовые сообщения.
4. Реклама и маркетинг: GIF-анимации эффективны для создания простых и запоминающихся рекламных материалов, таких как баннеры.

В целом, формат GIF-анимации остается популярным инструментом для создания кратких и занимательных анимированных изображений, подходящих для различных целей в онлайн-среде.

## **2.2 Сведения из теории**

Формат файла: GIF (Graphics Interchange Format)

Цветовая глубина: GIF поддерживает индексированный цвет с возможностью использования до 256 цветов из палитры 24-битного RGB. Каждый пиксель в GIF-анимации ссылается на индекс цвета в глобальной цветовой таблице.

Анимация: GIF-анимация представляет собой последовательность изображений (кадров), проигрываемых в определенной последовательности. Кадры могут иметь различные задержки, определяющие скорость воспроизведения.

Прозрачность: GIF поддерживает прозрачность, позволяя определить один цвет из палитры как прозрачный. Это особенно полезно для создания изображений с прозрачными фонами.

Циклическое воспроизведение: GIF-анимации могут быть настроены на циклическое воспроизведение, что означает повторение анимации после завершения последнего кадра.

Сжатие: Для сжатия изображений GIF использует алгоритм Lempel-Ziv-Welch (LZW), что делает формат эффективным для передачи через интернет.

Размер файла: GIF-анимации обычно имеют относительно небольшой размер файла благодаря использованию индексированной цветовой палитры и сжатию LZW.

Разрешение: GIF не ограничивает разрешение изображений, что позволяет создавать анимации различных размеров и форматов.

Поддержка множественных кадров: GIF может содержать неограниченное количество кадров, что предоставляет возможность создания как простых, так и сложных анимаций.

Применение: GIF-анимации часто используются в веб-дизайне, социальных сетях, рекламе и других областях, где требуется компактный формат с возможностью представления анимации.

Технические характеристики формата GIF делают его удобным средством для создания простых, но эффективных анимаций, легко распространяемых и отображаемых в различных средах.

## **2.3 Библиотеки, реализующие функцию создания GIF-анимации**

В языке программирования C# существует несколько библиотек, которые обеспечивают функциональность для работы с GIF-анимациями. Вот несколько таких библиотек:

1. ImageMagick:

* Описание: ImageMagick — мощная библиотека для обработки изображений, которая поддерживает множество форматов, включая GIF. Она предоставляет .NET-обертку, известную как Magick.NET, которая позволяет взаимодействовать с функциональностью ImageMagick в C#.
* Особенности:
  + - Создание и обработка GIF-анимаций.
    - Многообразие инструментов для манипуляции изображениями.

1. FreeImage:
   * Описание: FreeImage — библиотека для работы с изображениями, поддерживающая формат GIF. В C# существует обертка FreeImage.NET.
   * Особенности:
     1. Создание и редактирование GIF-анимаций.
     2. Поддержка различных операций с изображениями.
2. GifRenderer:

* Описание: GifRenderer — библиотека, специально разработанная для работы с GIF-анимациями в C#.
* Особенности:
  1. Простота в использовании.
  2. Поддержка создания и отображения GIF-анимаций.

1. ImageSharp:

* Описание: ImageSharp — библиотека, ориентированная на обработку изображений в .NET, включая формат GIF.
* Особенности:
  1. Обширный функционал для работы с изображениями.
  2. Возможность создания и манипуляции GIF-анимациями.

1. AnimatedGif:

* Описание: AnimatedGif — простая библиотека для создания GIF-анимаций в C#.
* Особенности:
  1. Легкая интеграция и использование.
  2. Базовые функции для создания анимаций.

При выборе библиотеки следует учитывать требования к проекту, уровень сложности задач, а также предпочтения разработчика. Большинство этих библиотек обладают активными сообществами и документацией, что облегчает их использование.

Для данной работы была выбрана такая библиотека, как BumpKit, как наиболее удобный, интуитивно понятный и эффективный инструмент.

BumpKit представляет собой библиотеку для языка программирования C#, которая расширяет возможности библиотеки System.Drawing GDI, часто используемой для работы с изображениями в .NET-приложениях. Она предназначена для удобства и эффективности в генерации изображений "на лету" в веб-приложениях, основанных на платформе .NET.

Основные возможности:

1. Масштабирование изображений:

* Метод System.Drawing.Image.ScaleToFit(...) позволяет изменять размер изображения с поддержкой масштабирования.
* Поддержка сохранения соотношения сторон и настройки размера относительно высоты или ширины.

1. Поворот изображений:
   * Метод System.Drawing.Image.Rotate(...) позволяет вращать изображения.
   * Поддержка центрирования изображения относительно границ и масштабирования для поддержания соотношения сторон.
2. Быстрый доступ к пикселям:
   * Метод System.Drawing.Image.CreateUnsafeContext() предоставляет доступ к пикселям изображения в безопасной среде.
   * Позволяет проводить быструю манипуляцию изображением с прямым доступом к памяти.
3. Генерация анимированных GIF:
   * Класс BumpKit.GifEncoder форматирует несколько изображений/кадров в единый анимированный GIF-файл.
   * Использует встроенное .NET-кодирование GIF (добавляет недостающие анимационные заголовки).
   * Относительно быстрый по сравнению с другими библиотеками для работы с GIF.
4. Эффекты шрифта:
   * Метод System.Drawing.Graphics.DrawString(...) обогащает функциональность DrawString, добавляя границу и эффект свечения шрифта.
   * Использует встроенную обработку шрифтов .NET.
   * Гибкая реализация API.

BumpKit предоставляет разработчикам удобные инструменты для манипуляции изображениями в .NET-приложениях, особенно в сфере генерации изображений для веб-приложений. Его функциональность включает в себя масштабирование, поворот, быстрый доступ к пикселям, генерацию анимированных GIF и эффекты шрифта.

## **2.4 Постановка задачи**

Задачи:

1. Рассмотреть теоретическую часть GIF-анимация
2. Выбрать расширение для создания GIF-анимация
3. Разработать программу по созданию GIF-анимация
4. Оформить интерфейс программы для интерактивного использования

Необходимо разработать программу, которая позволяет наиболее удобным и интуитивно понятным способом создавать GIF-анимации из набора изображений. Можно добавить функцию настройки скорость GIF-анимации для создания более плавной анимации, а также возможность сохранения по указанному пути.

# **3 Технология разработки программы**

## **3.1 Описание алгоритмов решения**

Метод Form1() – запуск интерактивного приложения и программы в целом.

Метод CreateGIF\_Click(object sender, EventArgs e) – основная функция программы, которая создает GIF-анимации из набора картинок.

Метод OnFormClosing(FormClosingEventArgs e) – перезапись метода закрытия программы, созданная с целью избегания ошибок, связанных с доступами к файлам.

Метод clearButton\_Click(object sender, EventArgs e) – очищает список ранее загруженных изображений.

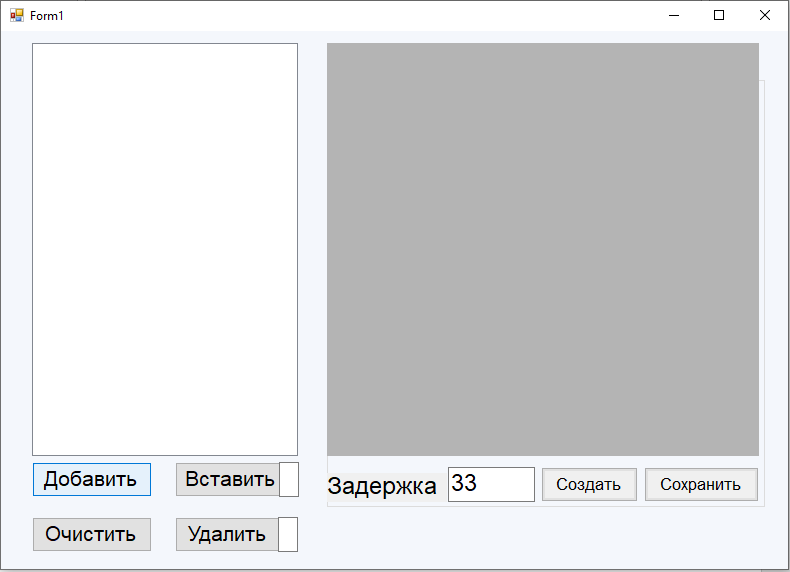
Метод saveButton\_Click(object sender, EventArgs e) – сохраняет GIF-анимацию по указанному пользователем пути.

Метод insertButton\_Click(object sender, EventArgs e) – метод был создан на случай, если пользователь нарушил порядок изображений в GIF-анимации, он позволяет вставить нужное изображение на определенную позицию.

Метод deleteButton\_Click(object sender, EventArgs e) – удаляет изображение из списка по номеру.

Метод OpenImage\_Click(object sender, EventArgs e) – добавляет изображение в конец списка.

## **3.2 Макет программы**



4

3

2

1

Рисунок 3.1 Приложение по созданию GIF-анимации.

Поле под цифрой 1 представляет собой список изображений, которые будут отображаться здесь же после их добавления.

Поле под цифрой 2 является областью, в которой можно посмотреть, как будет выглядеть GIF-анимация, не сохраняя её. Для этого нужно будет нажать на кнопку «Создать» из области 4, добавив перед этим изображения и, если нужно, поменяв в этом же поле задержку между кадрами.

В поле под цифрой 3 реализованы следующее кнопки:

* «Добавить» – метод OpenImage\_Click(object sender, EventArgs e);
* «Вставить» – метод insertButton\_Click(object sender, EventArgs e). Справа от кнопки находится поле, в котором необходимо указать позицию изображения, перед которым необходимо поместить изображение;
* «Очистить» – метод clearButton\_Click(object sender, EventArgs e);
* «Удалить» – метод deleteButton\_Click(object sender, EventArgs e). Справа от кнопки находится поле, в котором необходимо указать позицию изображения, которое необходимо удалить.

В поле под цифрой 4 находятся две кнопки: «Создать» (метод CreateGIF\_Click(object sender, EventArgs e)) и «Сохранить» (saveButton\_Click(object sender, EventArgs e)); а также поле с подписью «Задержка», которое выставляет задержку между изображениями GIF-анимации в мс (по умолчанию 33).

## **3.3 Описание программы**

using BumpKit;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

namespace Kursach

{

public partial class Form1 : Form

{

public static List<string> imagesLinks = new List<string>();

ImageList imgList = new ImageList();

public Form1()

{

//Выбор размера отображаемых в списке изображений, только визуал

imgList.ImageSize = new Size(128, 128);

InitializeComponent();

}

protected override void OnFormClosing(FormClosingEventArgs e)

{

base.OnFormClosing(e);

//освобождаем память и удаляем дамп-файл

if (File.Exists("dump.gif"))

{

if (GIFPictureBox.Image != null)

GIFPictureBox.Image.Dispose();

File.Delete("dump.gif");

}

}

private void OpenImage\_Click(object sender, EventArgs e)

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.BMP;\*.PNG;\*.JPG)|\*.BMP;\*.PNG;\*.JPG|All files (\*.\*)|\*.\*";

if (open.ShowDialog() != System.Windows.Forms.DialogResult.OK) { return; }

//imagesLinks - ссылка на картинки img - лист изображений

imagesLinks.Add(open.FileName);

imgList.Images.Add(Image.FromFile(imagesLinks[imagesLinks.Count-1]));

listView1.Items.Add((imagesLinks.Count).ToString(), imagesLinks.Count - 1);

listView1.LargeImageList = imgList;

}

private void deleteButton\_Click(object sender, EventArgs e)

{

if (imagesLinks.Count == 0)

{

MessageBox.Show("Сначала введите первый элемент");

}

else if (Convert.ToInt32(textBox1.Text) > 0)

{

//imagesLinks - ссылка на картинки img - лист изображений

imgList.Images.Clear();

imagesLinks.RemoveAt(Convert.ToInt32(textBox1.Text)-1);

foreach (var item in imagesLinks)

imgList.Images.Add(Image.FromFile(item));

listView1.Items.Clear();

for (int i = 0; i < imagesLinks.Count; i++)

{

listView1.Items.Add((i + 1).ToString(), i);

}

listView1.LargeImageList = imgList;

}

else

{

MessageBox.Show("Введите значение в пределах диапазона от 1 до " + imagesLinks.Count);

}

}

private void insertButton\_Click(object sender, EventArgs e)

{

if (imagesLinks.Count == 0)

{

MessageBox.Show("Сначала введите первый элемент");

}

else if (Convert.ToInt32(textBox2.Text) < imagesLinks.Count && Convert.ToInt32(textBox2.Text) > 0)

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.BMP;\*.PNG;\*.JPG)|\*.BMP;\*.PNG;\*.JPG|All files (\*.\*)|\*.\*";

if (open.ShowDialog() != System.Windows.Forms.DialogResult.OK) { return; }

//imagesLinks - ссылка на картинки img - лист изображений

imgList.Images.Clear();

imagesLinks.Insert(Convert.ToInt32(textBox2.Text) - 1, open.FileName);

foreach (var item in imagesLinks)

imgList.Images.Add(Image.FromFile(item));

listView1.Items.Clear();

for (int i = 0; i < imagesLinks.Count; i++)

{

listView1.Items.Add((i + 1).ToString(), i);

}

listView1.LargeImageList = imgList;

}

else

{

MessageBox.Show("Введите значение в пределах диапазона от 1 до " + imagesLinks.Count);

}

}

private void CreateGIF\_Click(object sender, EventArgs e)

{

if (GIFPictureBox.Image != null)

{

GIFPictureBox.Image.Dispose();

}

if (imagesLinks.Count > 0)

{

double delay;

if (delayNum.Text != "")

{

delay = Convert.ToDouble(delayNum.Text);

}

else

{

delay = 33;

}

using (var gif = File.OpenWrite("dump.gif"))

{

using (var encoder = new GifEncoder(gif))

for (var i = 0; i < imagesLinks.Count; i += 1)

{

using (var frame = Image.FromFile(imagesLinks[i]))

{

encoder.FrameDelay = TimeSpan.FromMilliseconds(delay);

encoder.AddFrame(frame);

}

}

}

GIFPictureBox.Image = Image.FromFile("dump.gif");

}

else

{

MessageBox.Show("Выберите изображения!");

}

}

private void clearButton\_Click(object sender, EventArgs e)

{

listView1.Clear();

imgList.Dispose();

imagesLinks.Clear();

if (GIFPictureBox.Image != null)

{

GIFPictureBox.Image.Dispose();

GIFPictureBox.Image = null;

}

delayNum.Text = "33";

}

private void saveButton\_Click(object sender, EventArgs e)

{

SaveFileDialog save = new SaveFileDialog();

save.Filter = "Image Files(\*.GIF)|\*.GIF|All files (\*.\*)|\*.\*";

if (save.ShowDialog() != System.Windows.Forms.DialogResult.OK) { return; }

var img = Image.FromFile("dump.gif");

img.Save(save.FileName);

}

}

}

## **3.4 Блок схемы методов программы**

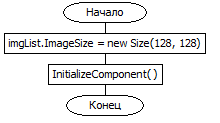


Рисунок 3.2 Блок схема метода Form1()

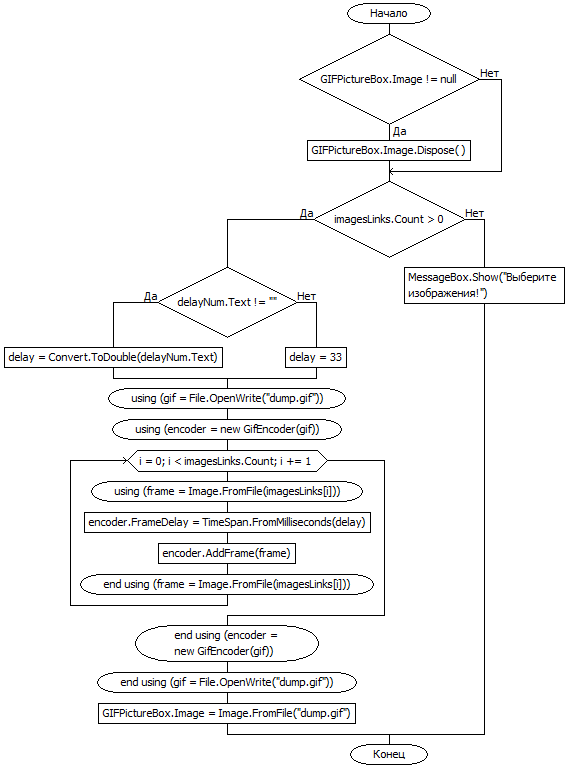


Рисунок 3.3 Блок схема метода CreateGIF\_Click(object sender, EventArgs e)

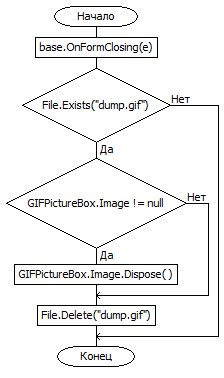


Рисунок 3.4 Блок схема метода OnFormClosing(FormClosingEventArgs e)

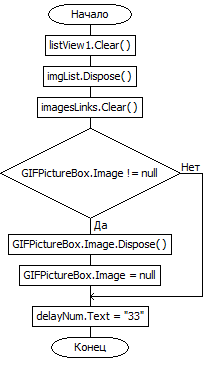


Рисунок 3.5 Блок схема метода clearButton\_Click(object sender, EventArgs e)

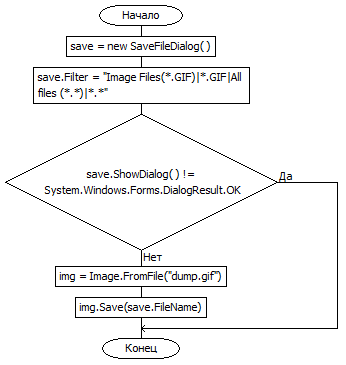


Рисунок 3.6 Блок схема метода saveButton\_Click(object sender, EventArgs e)

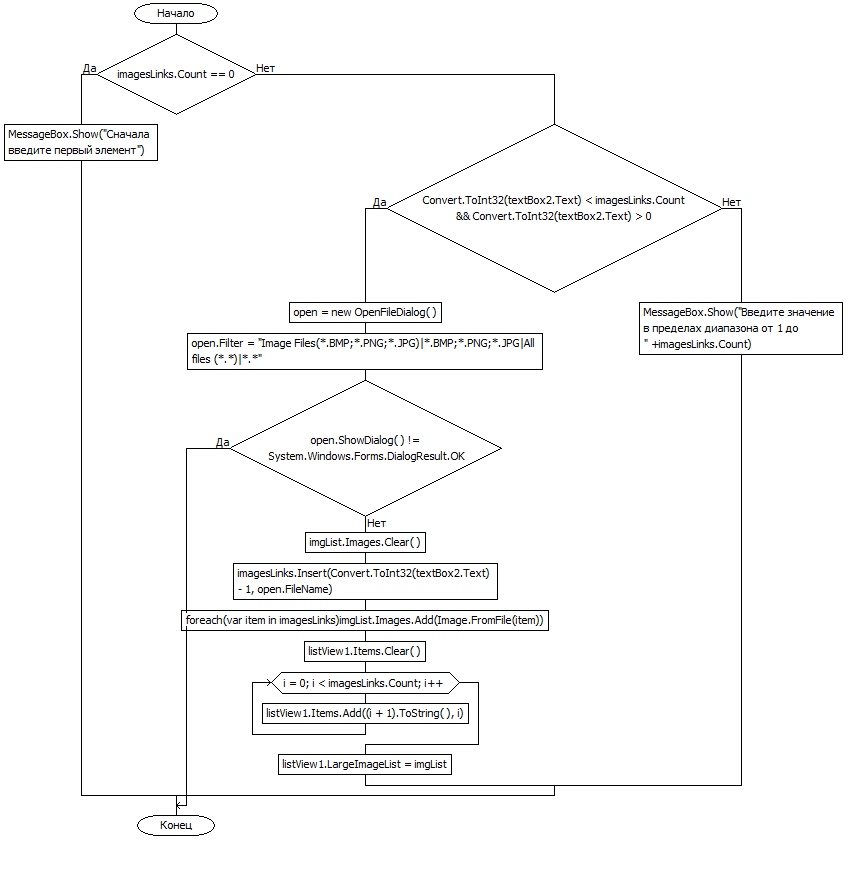


Рисунок 3.7 Блок схема метода insertButton\_Click(object sender, EventArgs e)

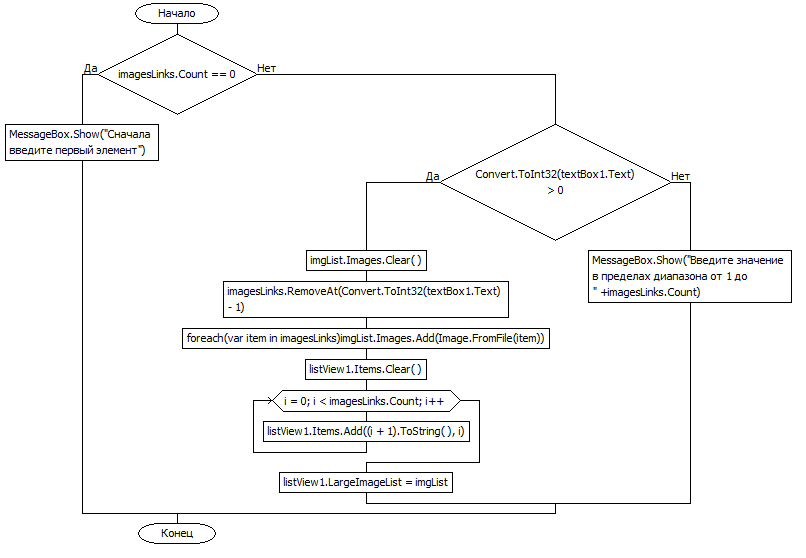


Рисунок 3.8 Блок схема метода deleteButton\_Click(object sender, EventArgs e)

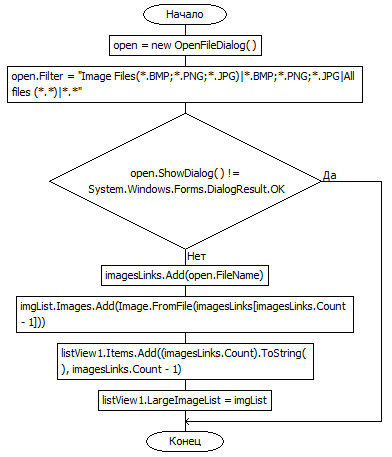


Рисунок 3.9 Блок схема метода OpenImage\_Click(object sender, EventArgs e)

## **3.5 Результат выполнения программы**

**Заключение**

В заключение выполненной работы можно отметить успешное достижение поставленных целей и решение поставленных задач. Работа была направлена на изучение теоретических аспектов создания GIF-анимаций, выбор подходящего расширения (в данном случае, библиотеки Bitmapkit в языке C#), разработку программы, обеспечивающей процесс создания GIF-анимаций из набора изображений.

Программа была успешно оформлена в виде интерактивного приложения, что обеспечивает удобство использования для конечного пользователя. Также был разработан небольшой мануал пользователя, который поможет пользователям разобраться с основными функциями приложения и провести процесс создания своих GIF-анимаций.

Оформление пояснительной записки позволяет ясно представить теоретический фон, выбранный подход к решению задачи, а также описание реализации программы. В ходе работы были учтены основные принципы создания GIF-анимаций, и разработанное приложение предоставляет простой и интуитивно понятный интерфейс для пользователей.

Таким образом, выполнение поставленных задач привело к созданию полноценного приложения, способного эффективно обрабатывать набор изображений и создавать качественные GIF-анимации. Результаты работы могут быть использованы как основа для дальнейших улучшений и развития данного приложения.

В ходе выполнения работы были получены знания не только о конструкциях языка, по которым были заданы задачи, но и о более сложных конструкциях языка, таких как выражения-предикаты, некоторые средства набора технологий LINQ (словари, кортежи, списки) и другие. Наиболее подробно были изучены классы System.Array и System.String.

**Список использованных источников**

* + - 1. Павловская Т.А. C#. Программирование на языке высокого уровня: Учебник для вузов. ‒ СПб.: Питер, 2014. ‒ 432 с.: ил.
      2. Марков В.Н. Алгоритмы и структуры данных: учеб. пособие/ В.Н. Марков. ‒ Краснодар: Изд. ФГБОУ ВО «КубГТУ», 2022. ‒ 207 с.
      3. Троелсен Эндрю, Джепикс Филипп Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд.: Пер. с англ. ‒ СПб.: ООО «Диалектика», 2018 ‒ 1328 с.: ил.
      4. https://awesomeopensource.com/project/Dawish/BitmapKit [Электронный ресурс] (дата обращение 01.11.2023).

**Приложение А. Тест на антиплагиат**

