Министерство образования Республики Беларусь

Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Программирование на языках высокого уровня

К ЗАЩИТЕ ДОПУСТИТЬ

Ассистент кафедры ЭВМ

А. В. Марзалюк

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе тему

МЕНЕДЖЕР ФОТОГРАФИЙ

на

БГУИР КП 1-40 02 01 413 ПЗ

Студент И. Р. Лагодич

Руководитель А.В.Марзалюк

МИНСК 2023

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

––"–––––––––––––––––20 г.

ЗАДАНИЕ

по курсовому проектированию

Студенту Лагодичу Илье Романовичу

1. Тема проекта Менеджер фотографий

2. Срок сдачи студентом законченного проекта 11 *декабря* 2023 *г.*

3. Исходные данные к проекту: нет.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

1. Лист задания.

2. Введение.

3. Постановка задачи

4. Обзор литературы.

4.1. Обзор методов и алгоритмов решения поставленной задачи.

5. Функциональное проектирование.

5.1. Структура входных и выходных данных.

5.2. Разработка диаграммы классов.

5.3. Описание классов.

6. Разработка программных модулей.

6.1. Разработка блок-схем алгоритмов (два наиболее важных метода).

6.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух

методов).

7. Результаты работы программы.

8. Заключение

9. Литература

10. Приложения

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

6. Консультант по проекту(с обозначением разделов проекта)

*Марзалюк А. В*.

7. Дата выдачи задания 15.09.*20*23 *г.*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

1. Выбор задания. Разработка содержания пояснительной

Перечень графического материала - к 01.10.2023 15 %

разделы 2, 3, 4 - к 01.10.2023 10 %:

раздел 5 - к 01.11.2023 20 %:

раздел б - к 01.12.2023 35 %:

разделы 7.89 - к 01.12.2023 5 %:

раздел 10 - к 01.12.2023 5 %:

оформление пояснительной записки и графического материала к 11.12.2023

-10 %:

*Защита курсового проекта с 2*1.12 *по 28*.12.2023 *г.*

РУКОВОДИТЕЛЬ– ––––––––– *Марзалюк А. В.*

Задание принял к исполнению –Х Лагодич *И. Р.*

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc153381038)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc153381039)

[2 ОБЗОР ЛИТЕРАТУРЫ 7](#_Toc153381040)

[2.1. Обзор методов и алгоритмов решения поставленной задач**и** 7](#_Toc153381041)

[3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 10](#_Toc153381042)

[3.1. Структура входных и выходных данных 10](#_Toc153381043)

[3.2. Разработка диаграммы классов 10](#_Toc153381044)

[3.3 Описание классов 10](#_Toc153381045)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 13](#_Toc153381046)

[4.1. Разработка блок-схем алгоритмов (два наиболее важных метода). 13](#_Toc153381047)

[4.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух 13](#_Toc153381048)

[4.2.1 Метод `drawPlot` класса `GraphDrawer`. 13](#_Toc153381049)

[4.2.2 Метод calculateFunction в классе Function. 14](#_Toc153381050)

[5. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ. 16](#_Toc153381051)

[ЗАКЛЮЧЕНИЕ 20](#_Toc153381052)

[ЛИТЕРАТУРА 21](#_Toc153381053)

[ПРИЛОЖЕНИЕ А 22](#_Toc153381054)

[ПРИЛОЖЕНИЕ Б 23](#_Toc153381055)

[ПРИЛОЖЕНИЕ В 24](#_Toc153381056)

[ПРИЛОЖЕНИЕ Г 25](#_Toc153381057)

# ВВЕДЕНИЕ

Эта работа посвящена разработке программного средства для менеджмента фотографий.

Целью работы является приобретение навыков разработки программных средств для визуализации, изменений и перемещений изображений на компьютере. В данной работе ставится задача создать удобное приложение, которое позволит пользователям перемещать и изменять изображения.

Программное средство будет разработано в виде графического приложения с использованием библиотеки Qt для создания современного пользовательского интерфейса, при этом используется язык программирования C++. Пользовательский интерфейс будет включать в себя следующие основные элементы:

1. Окно для просмотра выбранного изображения.

2. Окно со списком изображений в выбранной директории.

3. Кнопки для выбора и изменений изображений.

В ходе работы применяются различные парадигмы программирования, включая объектно-ориентированное программирование и работу с данными, чтобы обеспечить функциональность приложения.

Таким образом, целью этой работы является создание удобного и интуитивно понятного программного средства для менеджмента фотографий.

# 1 ПОСТАНОВКА ЗАДАЧИ

Разработать программное средство «менеджер фотографий.». Программа должна предоставлять удобный пользовательский интерфейс для открытия, просмотра и управлением изображениями. Задачи работы:

Задачи работы:

1. Изучить теорию по теме курсового проекта.
2. Изучить основные принципы работы с файлами и их изменениями.
3. Разработать пользовательский интерфейс программы, включающий в себя:
   * Окно для просмотра выбранного изображения.
   * Окно со списком изображений в выбранной директории.
   * Кнопки для выбора и изменений изображений.
4. Реализовать алгоритмы управления изображениями, в том числе: переименования и перемещения, на основе введенных пользователем данных.
5. Провести тестирование программы.
6. Оформить пояснительную записку.

В программном средстве должна быть реализовано обработка исключительных ситуаций.

# 2 ОБЗОР ЛИТЕРАТУРЫ

В данном разделе проведен обзор существующих методов и алгоритмов, используемых для менеджмента фотографий. Этот обзор поможет определить подходы, которые можно применить при разработке программного средства "Менеджер фотографий" на основе библиотеки Qt и языка программирования C++.

Фотография - технология записи изображения путём регистрации оптических излучений с помощью светочувствительного фотоматериала или полупроводникового преобразователя.

**Изображение, цифровое изображение** (англ. image - образ) - это один графический объект или картинка на экране монитора. Изображения создаются с помощью фотоаппаратов, сканеров, скриншотов, программ для создания и редактирования изображений, например, Paint и Photoshop.

Менеджер фотографий – это программа с широким функционалом, которая позволяет структурировать практически все форматы изображений, просматривать, редактировать, а также распределять их в любые, удобные для вас, категории и альбомы.

Примеры различных форматов изображений:

1. .bmp: Файлы BMP хранятся в виде 2D-изображений различных размеров, цветов и значений глубины цвета без сжатия данных, цветовых профилей или альфа-каналов. Они содержат информацию о пикселях изображения.

2. .png: PNG это растровый формат хранения графической информации, использующий сжатие без потерь по алгоритму Deflate.

3. .jpeg: JPEG это графический формат, применяемый для хранения изображений. Формат JPEG поддерживает, RGB (миллионы цветов) и шкалу серого. JPEG сохраняет всю информацию о цвете в изображении, однако сжимает размер файла путем выборочного удаления данных. Файлы, содержащие данные JPEG, обычно имеют расширения .jpg, .jfif, .jpe или .jpeg. Однако .jpg является самым популярным из них на всех платформах.

**2.1. Обзор методов и алгоритмов решения поставленной задачи**

1. Управление масштабом и интерактивность:

- Масштабирование: Для удобства пользователей важно предоставить возможность изменения масштаба. Это может быть реализовано путем изменения изображения по ширине и высоте.

- Интерактивные элементы: Добавление элементов управления, таких как кнопки для масштабирования, кнопки для переключения между изображениями, выбора папки для их открытия, перемещения, а также переименования.

2. Библиотека Qt и C++:

-Использование библиотеки Qt: Библиотека Qt предоставляет богатые средства для создания графических приложений с интерфейсами. Она включает в себя классы для рисования, обработки событий и создания пользовательских виджетов.

-Язык программирования C++: Программирование на C++ обеспечивает высокую производительность и контроль над памятью, что важно при разработке графических приложений.

3. Исследование аналогичных приложений:

-Приложения для менеджмента фотографий: Исследование существующих приложений, таких как IrfanView[1] (рисунок 2.1.1), XnView, FastStone и Google Photo, может дать представление о стандартах и ожиданиях пользователей.

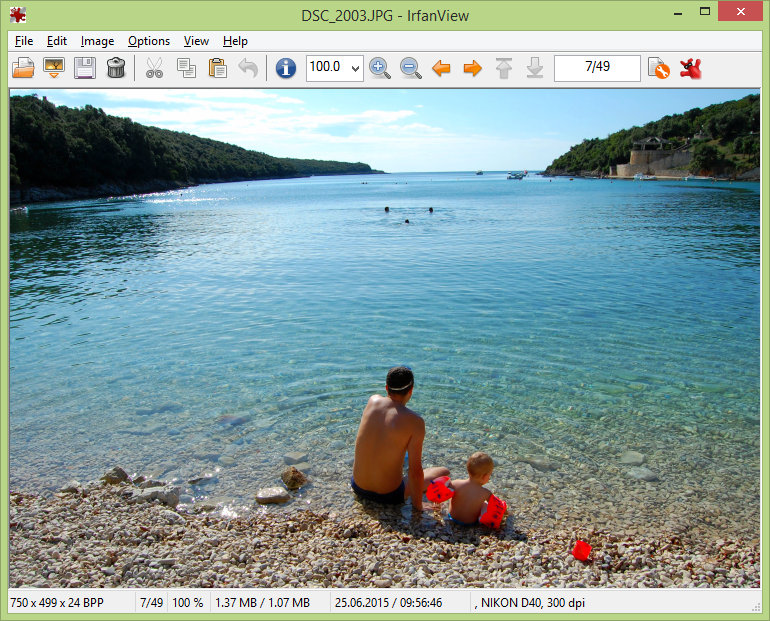


Рисунок 2.1.1 – приложение для построений графиков IrfanView

В заключение, обзор литературы показывает, что существуют разнообразные подходы для постройки пользовательского интерфейса, а также его функционала. Интерактивность и удобство использования играют важную роль в разработке подобных приложений. Применение библиотеки Qt и языка C++ предоставляет мощные инструменты для создания высококачественных приложений.

# 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В рамках данной главы рассмотренны основные аспекты функционального проектирования для менеджмента фотографий. Этот этап играет ключевую роль в создании программного продукта, так как определяет структуру входных и выходных данных, а также включает в себя разработку диаграммы классов и описание этих классов.

**3.1. Структура входных и выходных данных**

В качестве входных данных используется любой выбранный файл формата: .bmp ; .png ; .jpeg . Структура которого представлена в таблице 3.1.1.

Таблица 3.1.1 – структура файла изображения

|  |  |
| --- | --- |
| Заголовок | Указывает формат файла и содержит общие сведения о файле, в том числе, высоту и ширину изображения |
| Палитра | Таблица цветов, в соответствии с которой элементам изображения присваиваются значения RGB-троек |
| Данные изображения | Компоненты и образцы компонентов, формирующие изображение |

Эта структура данных позволяет увидеть название, разрешение (высоту и ширину), размер файла изображения, его теги и остальные компоненты.

**3.2. Разработка диаграммы классов**

Диаграммы классов являются важным инструментом в объектно-ориентированном программировании и разработке ПО. Вот почему они необходимы:

1. Помогают визуализировать структуру системы, отображая классы, их атрибуты и методы.
2. Облегчают понимание взаимосвязей между классами и пакетами.
3. Содействуют проектированию четких и эффективных архитектурных решений.
4. Позволяют представить базовые отношения и взаимодействия между классами.
5. Создают основу для более глубокого анализа и проектирования системы.

Диаграмма классов представлена в приложении А.

**3.3 Описание классов**

Класс Imageitem – класс для представления базовых параметров изображения и изменения его размера.

Конструктор:

Imageitem(QWidget \*parent = nullptr) – Выделяет память для переменных класса и компонует их в виджете.

Методы:

- setData(QPixmap Image, QString TitleText, QString InfoText, QString Path) – устанавливает изображение, его название, информацию о размере файла и его путь.

- getPath() - возвращает путь изображения.

Приватные поля:

- img - изображение.

- title – название изображения.

- info – информация о размере изображения.

- path – путь к изображению.

Класс Mainwindow – класс для представления изображений и взаимодействия с ними.

Конструктор:

MainWindow(QWidget \*parent = nullptr):

- Создает интерфейс и выделяет память для ярлыка изображения.

Приватные слоты.

- on\_action\_Open\_File\_triggered() – Выбор изображения для открытия.

- setImagesList(QString FileName) – Добавляет изображения в список для их просмотра из указанной директории.

- setOrigin(QString path) – Сохраняет название и указанный путь к изображению.

- on\_listWidget\_itemDoubleClicked(QListWdigetItem \* item) – Устанавливает выбранное изображение из списка в активное окно для просмотра и взаимодействия с ним.

- on\_action\_Zoom\_in\_triggered() – Приближает выбранное изображение умножив текущий размер на 1.25.

- on\_action\_Zoom\_out\_triggered() – Отдаляет выбранное изображение умножив текущий размер на 0.75.

- on\_action\_Original\_size\_triggered() – Устанавливает оригинальный размер изображения в активном окне.

- on\_action\_Adjust\_the\_size\_triggered() – Подгоняет размер выбранного изображения по вертикали/горизонтали(в зависимости от изображения) под размер активного окна.

- on\_actionSave\_File\_triggered() – Сохраняет выбранное изображение.

- on\_actionChoose\_directory\_to\_move\_this\_photo\_triggered() – Выбор директории для перемещения изображений.

- on\_Move\_Photo\_Button\_clicked() – Перемещает изображение в выбранную директорию

- on\_Rename\_Photo\_Button\_clicked() – Переименовывает изображение в написанное пользователем название.

Приватные поля:

- imageLabel – Ярлык изображения.

- fileDir – Название директории с выбранным изображением.

- path – Путь к изображению.

- fileName – Название файла изображения.

- destName – Выбранная директория для перемещения изображений.

# 4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

В данном разделе описана разработка методов приложения, которые являются ключевыми для функционирования данного приложения по менеджменту фотографий. Более детально были рассмотрены два наиболее значимых метода, а именно setImagesList и on\_listWidget\_itemDoubleClicked, а также представлена шаг за шагом их реализация.

Передо мной стояла задача создать эффективные алгоритмы, способные дать пользователю возможность в просмотре сразу нескольких фотографий в списке и их выборе для дальнейшего взаимодействия с ними. Эти программные модули стали ключевыми строительными блоками моего приложения, определяя его функциональность.

**4.1 Разработка блок-схем алгоритмов (два наиболее важных метода)**

Метод setImagesList в классе Mainwindow предназначен для размещения изображений в списке для их просмотра. Блок-схема данного алгортима предстаавлена в приложении Б.

Метод on\_listWidget\_itemDoubleClicked в классе Mainwindow предназначен для открытия для просмотра и становления выбранного изображения активным. Блок-схема данного алгортима предстаавлена в приложении В.

**4.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух методов)**

**4.2.1 Метод ` setImagesList ` класса ` Mainwindow ` отвечает за добавление изображений в список в виджете `listWidget`**

Входные данные: FileName (Путь к выбранному изображению).

Выходные данные: нет.

1. Начало

2. Очистить список в listWidget.

3. Инициализация переменных:

- QFileInfo info: Информация о файле изображения.

- QDir dir: Путь к изображению.

- QStringList filtr: Фильтр форматов изображений.

- QFileInfoList list: Список отфильтрованных изображений.

- QPixmap pix: Изображение.

- QFileInfo f: Аргумент для данных списка.

- int w: Ширина изображения.

- int h: Высота изображения.

- double fSize: Размер изображения в байтах.

- QString Info: Строка с информацией о разрешении и размере изображения.

- QListWidgetItem \*item: Указатель на ячейку изображения в списке.

- Imageitem \*imgitem: Объект класса, хранящий само изображение и информацию о нем.

4. Создание списка файлов с отфильтрованными форматами изображений.

5. Очистка активного окна и выход из функции, если список пуст:

6. Установка первого в списке изображения в активное окно, если ранее оно не было установлено.

7. Проход по изображениям в списке.

8. Получение пути к файлу и его изображения.

9. Получение информации о разрешении изображения и его размере в килобайтах.

10. Запись разрешения и размера изображенияя в строку Info.

11. Создание ячейки в listWidget.

12. Создание объекта imgitem класса Imageitem.

13. Вызов метода setData объекта imgitem, для заполнения полученной информации о изображении.

14. Изменение размера ячейки под кол-во информации о изображении.

15. Добавление мини-изображения и информации о нем в ячейку списка.

16. Если изображение в списке было не последним, то идет переход к следующему изображению и возвращение к шагу 7.

17. Конец

Этот метод обеспечивает заполнение и отображение в списке всех изображений форматов: .bmp, .png, .jpeg с информацией о них.

**4.2.2 Метод on\_listWidget\_itemDoubleClicked в классе Mainwindow.**

Входные данные: QListWidgetItem \*item (Указатель на выбранную ячейку виджета listWidget).

Выходные данные: нет.

1. Начало

2. Инициализация переменных:

- Imageitem \*imgitem: Объект класса, хранящий само изображение и информацию о нем.

- QPixmap pix: Изображение.

3. Заполнение информации объекта imgitem из ячейки виджета, путем приведения ячейки к типу Imageitem\*.

4. Получение изображения из ячейки.

5. Установка изображения в активное окно.

6. Вызов функции setOrigin() для сохранения информации о название и пути к изобраежнию.

7. Конец

Этот метод позволяет установить выбранное изображение из списка в активное окно, для дальнейшего взаимодействия с ним.

# 5 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

В рамках данной работы был разработан итерфейс программы представленный на рисунке 5.1.

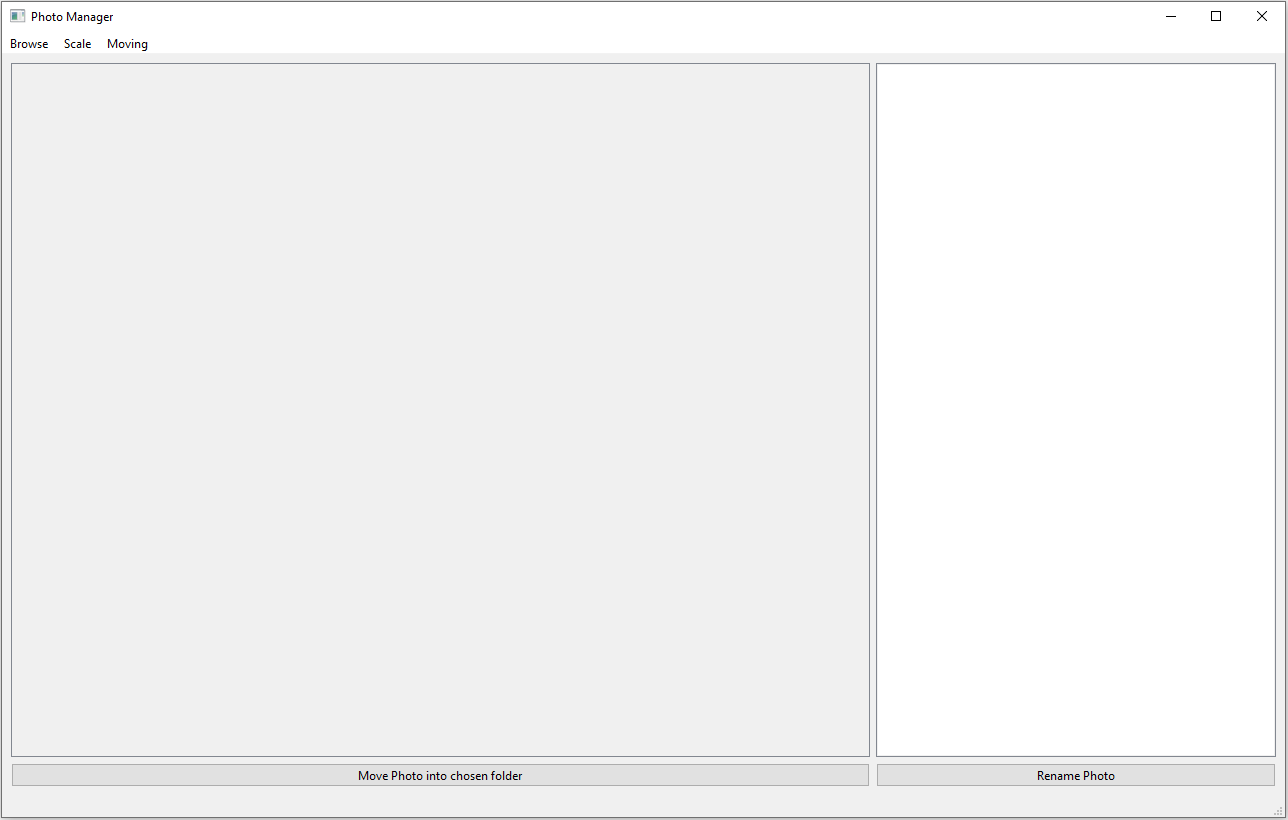


Рисунок 5.1 – Интрефейс программы

На интерфейсе представленно:

* Активное окно с выбранным изборажением.
* Список изображений в выбранной директории:
* Кнопка “Move Photo into chosen folder” по нажатию которой происходит перемещение выбранного изображения в указанную пользователем папку.
* Кнопка “Rename Photo” по нажатию которой пользователю открывается окно для ввода нового названия изображения, если это возможно

Интерфейс программы при открытой директории представлен на рисунке 5.2.

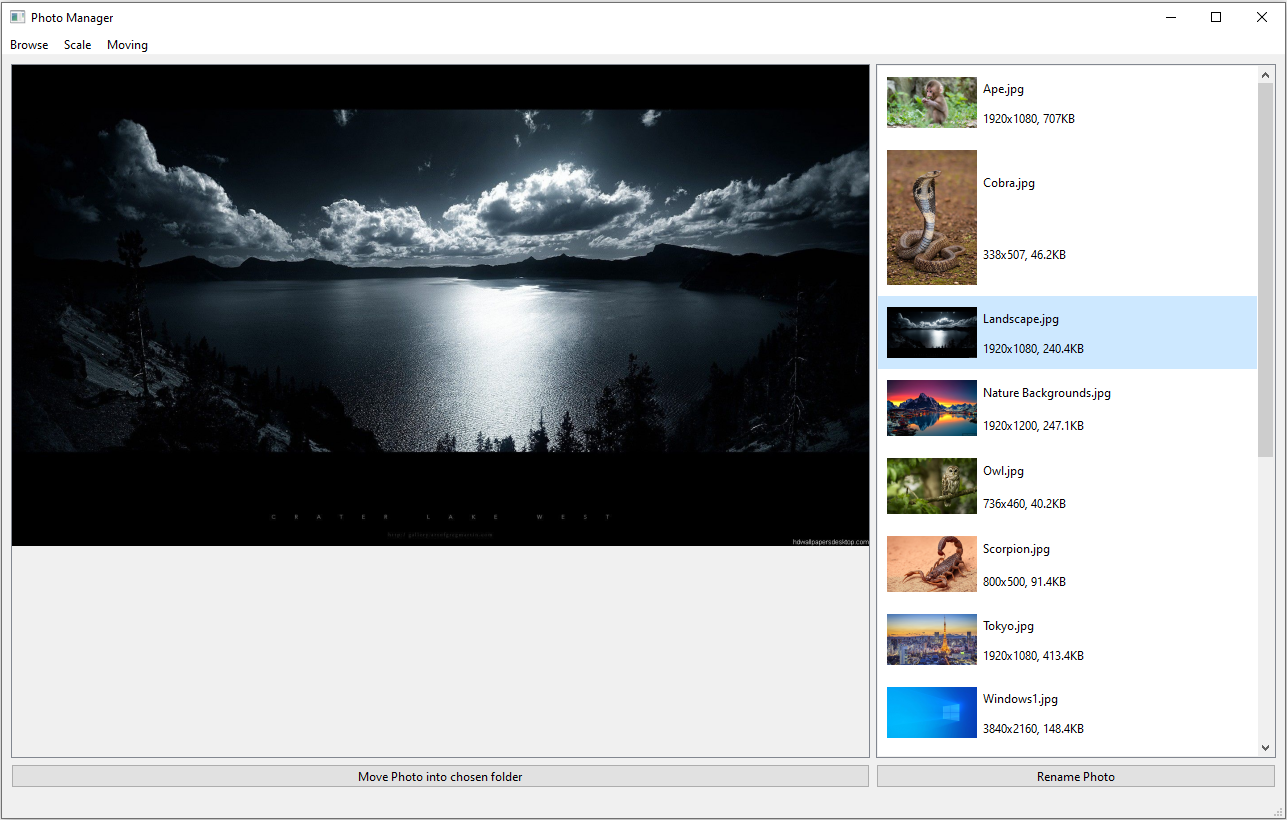


Рисунок 5.2 – Открытая директория

Переключение между изображениями в списке. (рис 5.3)

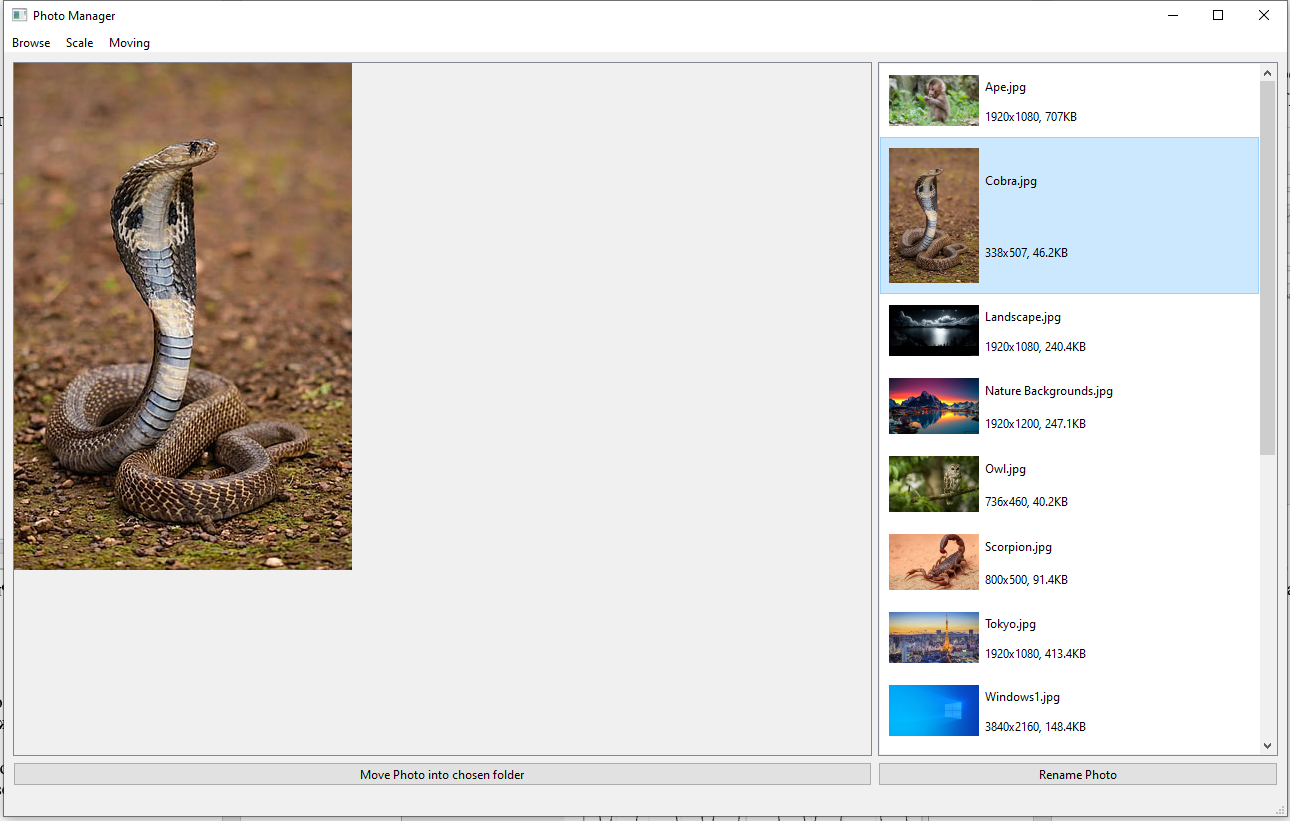


Рисунок 5.3 – Двойной клик по ячейке

При переходе по “Scale 🡪 Adjust the size” подгоняется размер активного изображения под размер окна: до (рис. 5.4.1) и после (рис. 5.4.2).

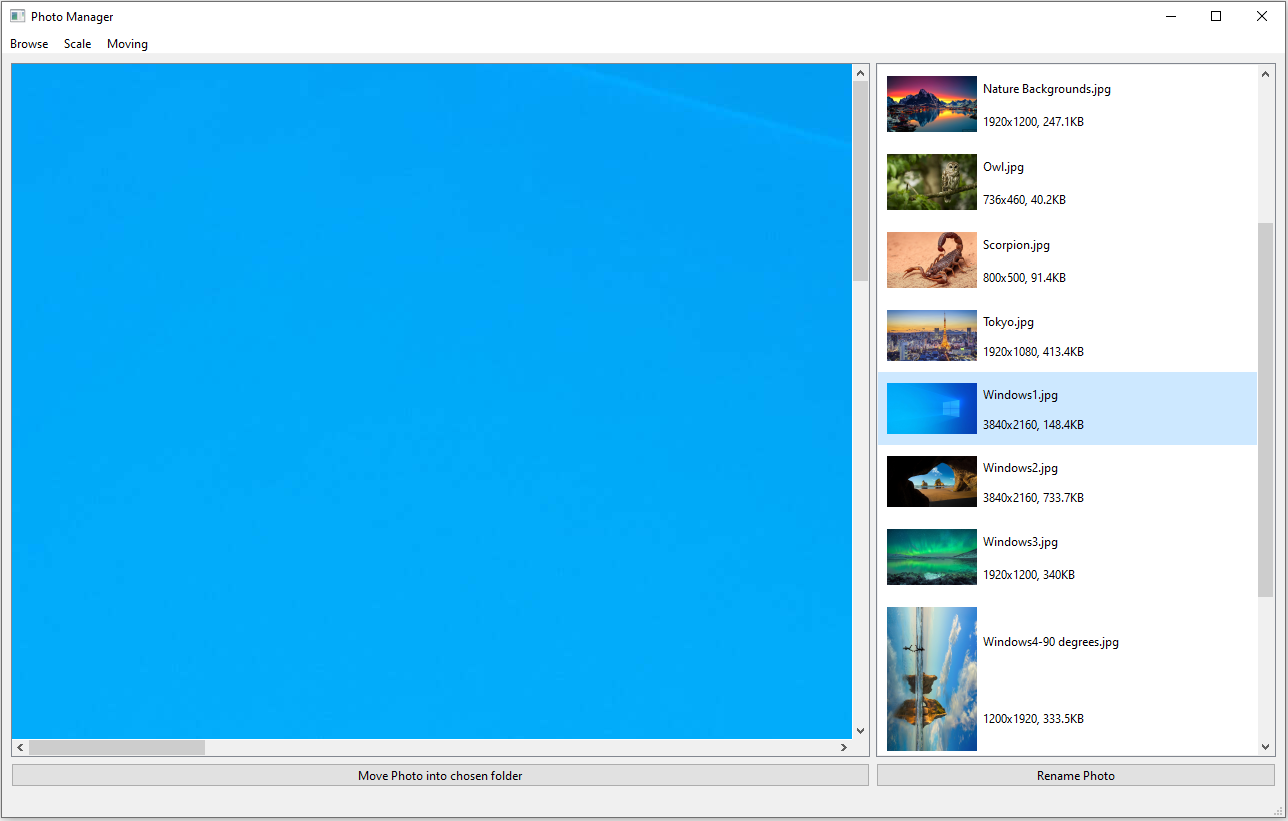


Рисунок 5.4.1 – Активное окно до подгона изборажения

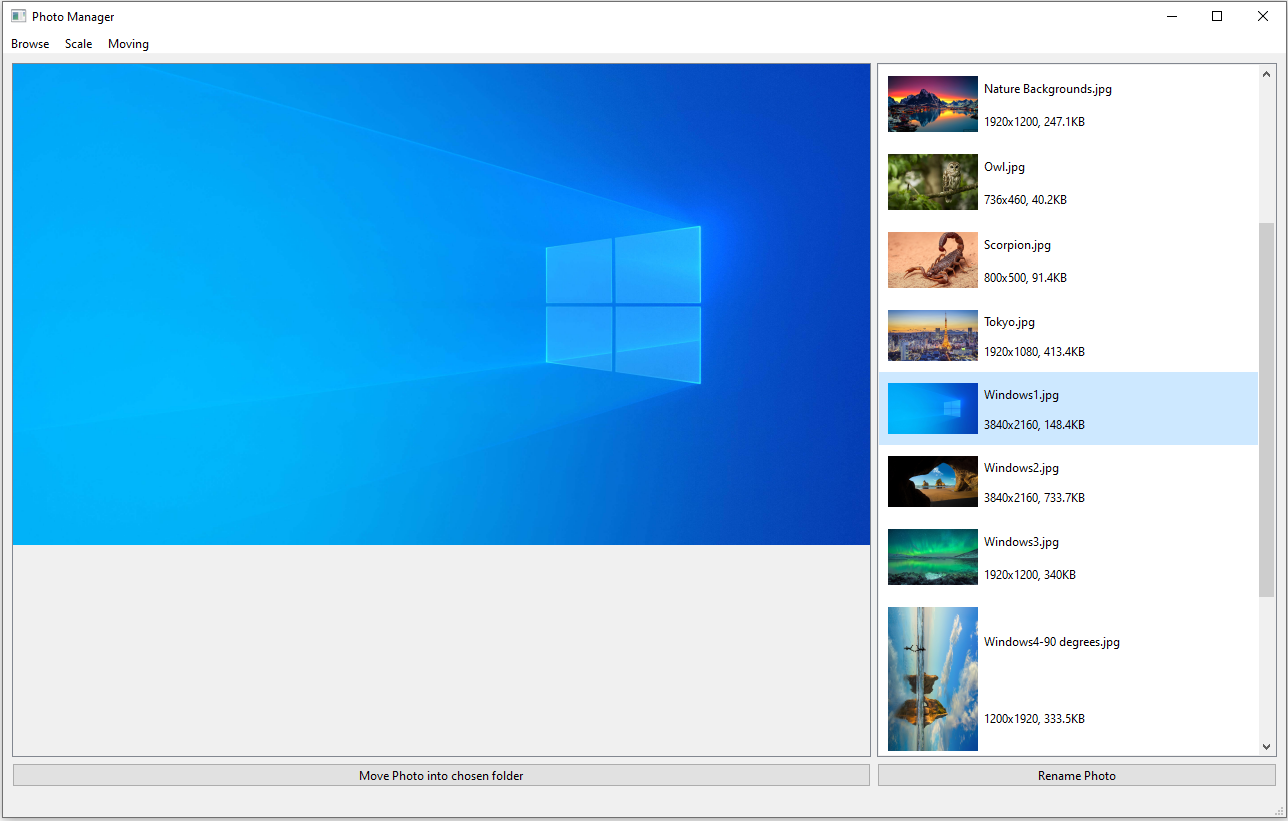


Рисунок 5.4.2 – Активное окно после подгона изборажения

Переименование изображения при нажатии на кнопку “Rename Photo” (рис. 5.5).

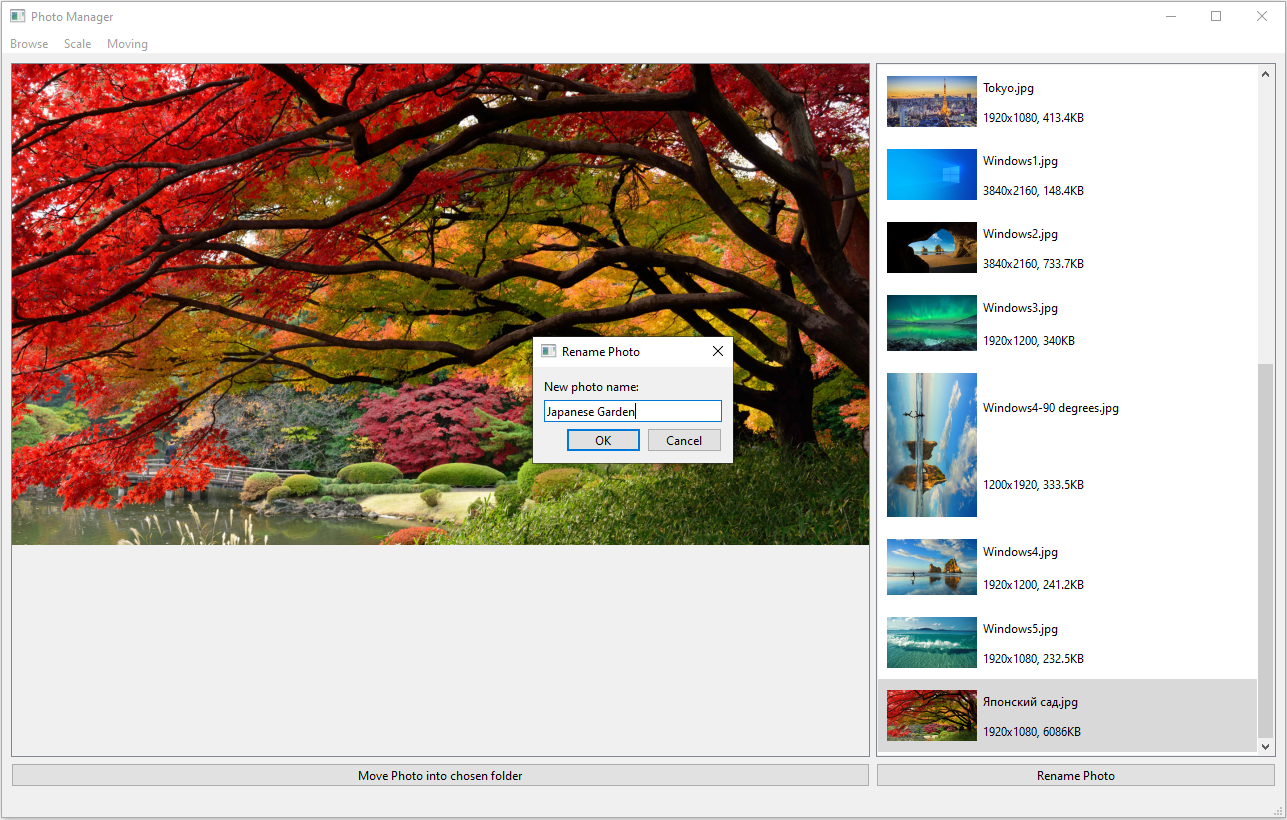


Рисунок 5.5 – Кнопка “Rename Photo”

# ЗАКЛЮЧЕНИЕ

В заключение проекта подчеркну важность достигнутых результатов и усилий, вложенных в разработку приложения для менджмента фотографий. В ходе работы было достигнуто несколько ключевых моментов:

1. Функциональность и удобство использования:

- Создано понятное приложение с простым и понятным интерфейсом, что делает управление фотографиями и их просмотр доступным даже для пользователей без специальных навыков.

2. Стабильность и безопасность:

- Реализованы механизмы обработки ошибок, что обеспечивает стабильную работу программы и предотвращает возможные проблемы при работе с фотографийями.

3. Масштабирование и адаптация:

- Разработан механизм масштабирования и автоматического подгона размера изображения под размер активного окна, учитывающий форму и размер виджета и изображения, что обеспечивает корректный и комфортный просмотр изображения

4. Работа с файлами

- Разработана удобная возможность перемещения и переименования изображений в выбранной директории с обновлением списка, после любой из данных операций.

Этот проект был довольно интересным в создании, и результаты, достигнутые в ходе работы, приносят удовлетворение. Программа готова к использованию и может быть полезной для тех, кто хочет работать с изображениями.

# ЛИТЕРАТУРА

[1] IrfanView [Электронный ресурс] // Приложение для просмотра и редактирования фотографий. [Режим доступа]: https://www.irfanview.com/.. – Дата доступа: 25.09.2023

[2] Бьерн Страуструп. Язык программирования С++/ . Пер. с англ. – М.: «Издательство БИНОМ», 2004. – 1098 с.

[3] Скляров В.А. Язык С++ и объектно-ориентированное программирование/. – Мн.: Выш.шк., 1997г. – 478 с.: ил.

[4] Луцик Ю.А., Ковальчук А.М., Лукьянова И.В. Объектноориентированное программирование на языке С++/. – М.: БГУИР, 2003. – 203 с.: ил.

# ПРИЛОЖЕНИЕ А

**(**обязательное**)**

Диаграмма классов

# ПРИЛОЖЕНИЕ Б

**(**обязательное**)**

Схема метода setImagesList

# ПРИЛОЖЕНИЕ В

**(**обязательное**)**

Схема метода on\_listWidget\_itemDoubleClicked

# ПРИЛОЖЕНИЕ Г

(обязательное)

Исходный текст программы