МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Документация

по лабораторной работе №2

Петрова Егора Андреевича

студента 2 курса, 10 группы

специальность «Прикладная информатика»

Руководитель практики:

Старший преподаватель кафедры компьютерных технологий и систем

С. С. Соловей

Минск, 2025

# Содержание

[Содержание 2](#_Toc210861938)

[Предисловие 4](#_Toc210861939)

[Часть 1 Общее описание 5](#_Toc210861940)

[1.1 Постановка задачи Написать приложение, считывающее из графического файла/файлов основную информацию об изображении. 5](#_Toc210861941)

[1.2 Основные возможности 5](#_Toc210861951)

[Глава 2 Руководство пользователя 6](#_Toc210861952)

[2.1 Запуск приложения 6](#_Toc210861953)

[2.2 Основные элементы интерфейса 6](#_Toc210861954)

[Глава 3 Устройство и архитектура кода приложения 9](#_Toc210861955)

[3.1 Структура проекта 9](#_Toc210861956)

[3.2 Взаимодействие компонентов 9](#_Toc210861957)

[3.3 Разбор кода 9](#_Toc210861967)

# Предисловие

Данное приложение было разработано как инструмент для анализа графических файлов различных форматов. Оно предоставляет комплексное решение для получения детальной информации об изображениях, что будет полезно для всех, кто работает с цифровой графикой.

Особенность "Анализатора графических файлов" заключается в его способности обрабатывать множество форматов изображений и предоставлять подробную техническую информацию о каждом файле. Приложение позволяет быстро анализировать как отдельные файлы, так и целые папки с изображениями, что значительно упрощает работу с большими коллекциями графических материалов.

Мы надеемся, что это приложение станет для вас надежным помощником в работе с графическими файлами, будь то профессиональные задачи, образовательные цели или личные проекты. Данная документация призвана помочь вам максимально эффективно использовать все возможности программы.

# Часть 1 Общее описание

## Постановка задачи Написать приложение, считывающее из графического файла/файлов основную информацию об изображении.

## Обязательно надо отображать следующее:

## имя файла;

## размер изображения (в пикселях);

## разрешение (dot/inch);

## глубина цвета;

## сжатие (для тех типов файлов, в которых оно может быть задано).

## 

## На проверку сдаются:

## exe-файл, который должен работать на любом ПК под Windows

## сопроводительная документация.

## Основные возможности

"Анализатор графических файлов" - это приложение для получения детальной технической информации о графических файлах.

**Программа позволяет:**

* **Анализировать различные форматы**: JPEG, PNG, GIF, BMP
* **Получать информацию о размерах**: Ширина, высота в пикселях
* **Определять разрешение**: DPI для печати
* **Анализировать глубину цвета**: Битовую глубину и цветовые модели
* **Определять тип сжатия**: С потерями/без потерь
* **Получать дополнительную информацию**: Размер файла, прозрачность, анимация, цветовые модели
* **Обрабатывать пакеты файлов**: Анализ нескольких файлов одновременно
* **Визуализировать данные**: Удобное табличное представление с подсказками

**Поддерживаемые форматы:**

* **JPEG/JPG** - формат с потерями для фотографий
* **PNG** - формат без потерь с поддержкой прозрачности
* **GIF** - формат с анимацией и индексированными цветами
* **BMP** - несжатый формат Windows
* **TIFF/TIF** - профессиональный формат с различными типами сжатия
* **PCX** - устаревший формат с сжатием RLE

# Глава 2 Руководство пользователя

## 2.1 Запуск приложения

* Через exe-файл на любом компьютере под Windows
* Запуск через QtCreator с исходным кодом

## 2.2 Основные элементы интерфейса

#### **Панель управления**

* Расположена в верхней части окна
* Содержит кнопки для выбора файлов и управления анализом
* Поле отображения текущего пути или количества файлов

#### **Список файлов**

* Расположен в левой части окна
* Отображает имена выбранных для анализа файлов
* Позволяет видеть общее количество файлов

#### **Таблица результатов**

* Расположена в основной части окна
* Содержит подробную информацию о каждом файле
* Колонки с техническими характеристиками изображений

#### **Прогресс-бар**

* Отображает ход анализа при обработке множества файлов
* Показывает текущий прогресс и общее количество файлов

#### **Строка состояния**

* Отображает информационные сообщения и статус операции
* Покажает количество найденных и обработанных файлов
  1. **Работа с приложением**

#### **Выбор файлов для анализа:**

* **Через выбор папки:**
  + Нажмите кнопку "Выбрать папку..."
  + Выберите папку, содержащую изображения
  + Приложение автоматически найдет все поддерживаемые форматы
* **Через выбор файлов:**
  + Нажмите кнопку "Выбрать файлы..."
  + Выберите один или несколько файлов в диалоговом окне
  + Поддерживается множественный выбор

#### **Анализ изображений:**

* Нажмите кнопку "Анализировать" после выбора файлов
* Наблюдайте за прогрессом в прогресс-баре
* Просматривайте результаты в таблице

**Имя файла**

* Отображает имя файла
* При наведении показывает preview изображения и полный путь

**Размер (px)**

* Ширина и высота изображения в пикселях
* Основной параметр для веб и цифрового использования

**Разрешение (DPI)**

* Количество точек на дюйм
* Важный параметр для печати
* Определяет физический размер при печати

**Глубина цвета**

* Количество бит на пиксель
* Определяет количество возможных цветов
* Типы: монохром, индексированные цвета, True Color

**Сжатие**

* Алгоритм сжатия изображения
* Типы: с потерями (JPEG), без потерь (PNG, GIF), без сжатия (BMP)

**Дополнительная информация**

* Размер файла на диске
* Поддержка прозрачности
* Цветовая модель (цветное/градации серого)
* Поддержка анимации
* Количество цветов в палитре (для GIF)

#### **Особенности работы**

* При анализе больших коллекций прогресс-бар показывает ход выполнения
* Подсказки при наведении на заголовки колонок объясняют технические термины
* Автоматическое определение характеристик для каждого формата
* Обработка ошибок для неподдерживаемых или поврежденных файлов

# Глава 3 Устройство и архитектура кода приложения

## 3.1 Структура проекта

Приложение построено с использованием фреймворка Qt и следует паттерну Model-View:

Анализатор графических файлов/

├── main.cpp # Точка входа

└── imageanalyzer.h/cpp # Основной класс (View + Controller + Model)

## 3.2 Взаимодействие компонентов

## Пользовательский интерфейс (ImageAnalyzer)

## ↑

## | Прямые вызовы методов

## ↓

## Анализ изображений (QImageReader)

## ↑

## | Чтение метаданных

## ↓

## Графические файлы (различные форматы)

## 3.3 Разбор кода

* + 1. **main.cpp:**

#include <QApplication>

#include "imageanalyzer.h"

int main(int argc, char \*argv[])

{

QApplication app(argc, argv);

app.setApplicationName("Image Analyzer");

ImageAnalyzer analyzer;

analyzer.show();

return app.exec();

}

* + 1. **Главный класс (ImageAnalyzer):**

**Конструктор и инициализация интерфейса:**

ImageAnalyzer::ImageAnalyzer(QWidget \*parent)

: QMainWindow(parent)

{

QWidget \*centralWidget = new QWidget(this);

setCentralWidget(centralWidget);

QVBoxLayout \*mainLayout = new QVBoxLayout(centralWidget);

// Создание элементов интерфейса

selectFolderButton = new QPushButton("Выбрать папку...", this);

selectFilesButton = new QPushButton("Выбрать файлы...", this);

clearButton = new QPushButton("Очистить список", this);

analyzeButton = new QPushButton("Анализировать", this);

// Настройка таблицы

table = new QTableWidget(this);

table->setColumnCount(6);

table->setHorizontalHeaderLabels(QStringList() << "Имя файла" << "Размер (px)" << "Разрешение (DPI)" << "Глубина цвета" << "Сжатие" << "Дополнительная информация");

// Подключение сигналов

connect(selectFolderButton, SIGNAL(clicked()), this, SLOT(selectFolder()));

connect(selectFilesButton, SIGNAL(clicked()), this, SLOT(selectFiles()));

connect(clearButton, SIGNAL(clicked()), this, SLOT(clearFiles()));

connect(analyzeButton, SIGNAL(clicked()), this, SLOT(analyzeImages()));

}Механизм обновления:

void ColorModel::update()

{

emit colorChanged(color()); // Сигнал об изменении цвета

emit rgbChanged(m\_rgb); // Сигнал об изменении RGB

emit hsvChanged(m\_hsv); // Сигнал об изменении HSV

emit xyzChanged(m\_xyz); // Сигнал об изменении XYZ

}

**Методы анализа изображений:**

void ImageAnalyzer::analyzeImage(const QString &filePath)

{

QFileInfo fileInfo(filePath);

QString fileName = fileInfo.fileName();

QString format = fileInfo.suffix().toUpper();

QImageReader reader(filePath);

if (!reader.canRead()) {

// Обработка ошибок чтения

return;

}

QSize size = reader.size();

QImage image;

if (!reader.read(&image)) {

// Обработка ошибок загрузки

return;

}

// Получение DPI

int dpi = (image.dotsPerMeterX() > 0 && image.dotsPerMeterY() > 0) ?

image.dotsPerMeterX() / 39.3701 : 0;

// Добавление данных в таблицу

int row = table->rowCount();

table->insertRow(row);

table->setItem(row, 0, new QTableWidgetItem(fileName));

table->setItem(row, 1, new QTableWidgetItem(QString("%1 x %2").arg(size.width()).arg(size.height())));

table->setItem(row, 2, new QTableWidgetItem(QString("%1").arg(dpi)));

table->setItem(row, 3, new QTableWidgetItem(getColorDepthInfo(image, format)));

table->setItem(row, 4, new QTableWidgetItem(getCompressionInfo(format)));

table->setItem(row, 5, new QTableWidgetItem(getExtraInfo(reader, image, format, fileInfo.size())));

}