Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г.

Короленко.»

**ОТЧЕТ**

**По учебной практике**

**ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем**

**Специальность 09.02.03 Программирование в компьютерных системах**

Выполнил:

Студент ОП СПО группа 33

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

Руководитель практики: ст. преподаватель кафедры

Математики и информатики Касаткин К.А

Г. Глазов 2018

Оглавление

[Введение 3](#_Toc9593811)

[Цель разработки 4](#_Toc9593812)

[Обзор аналогов 5](#_Toc9593813)

[Постановка задачи и требования, предъявляемые к программе 10](#_Toc9593814)

[Описание алгоритма работы программы 11](#_Toc9593815)

[Выбор и обоснование выбора средства разработки, исходный код 14](#_Toc9593816)

[Тестирование программы 20](#_Toc9593817)

[Руководство пользователя 21](#_Toc9593818)

[Ссылки на интернет-ресурсы 22](#_Toc9593819)

### Введение

Целью прохождения учебной практики было повышение личных навыков программирования, путем создания программного обеспечения для компьютерных систем.

### Цель разработки

Целью разработки было создание программного обеспечения, в моем случае это “Фотогалерея с выводом метаданных”. Функции:

* Загрузка изображения
* Вывод изображения
* Вывод метаданных
* Масштабирование (вручную)

### Обзор аналогов

Среди программ для просмотра изображений есть 6 которые больше всех пользуются популярностью:

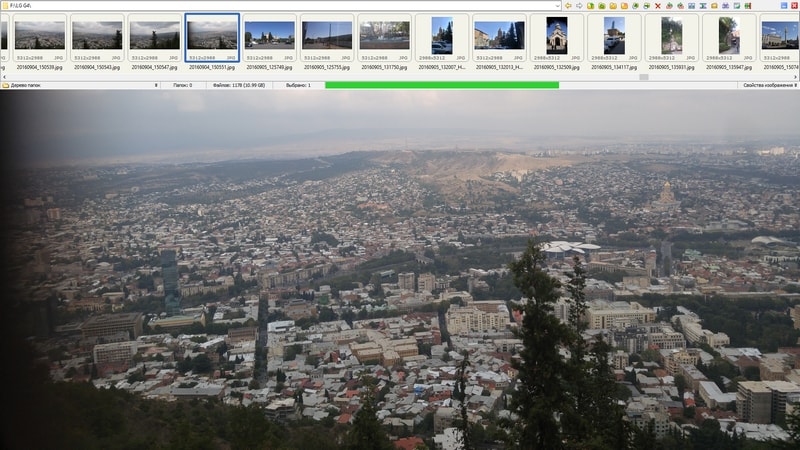
* FastStone Image Viewer
* ACDSee
* IrfanView
* XnView
* STDU Viewer
* Simply Slideshow

Перейдем к их более подробному рассмотрению.

FastStone Image Viewer

Если верить отзывам пользователей и соотношению позитива и негатива, FastStone – лучший просмотрщик фото наших дней.

Программа работает быстро, имеет встроенный редактор, умеет конвертировать фото в пакетном режиме. Отметим необычный дизайн интерфейса: в полноэкранном режиме достаточно завести курсор за край дисплея, чтобы получить доступ к различным разделам меню.

[](https://compconfig.ru/wp-content/uploads/2018/07/vybor-foto-v-FastStone-Image-Viewer.jpg)

FastStone поддерживает множество форматов изображений – от массовых до профессиональных.

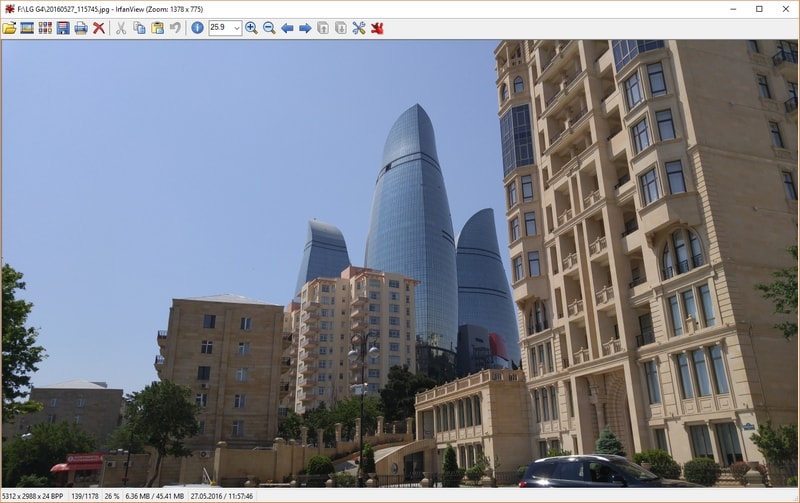
ACDSee

ACDSee — коммерческая (в прошлом — условно-бесплатная) программа для просмотра и управления коллекцией изображений, выпускаемая ACD Systems. Содержит многочисленные инструменты для обработки изображений, в том числе и пакетной. В последних версиях имеет два режима просмотра: быстрый, в котором доступны только инструменты поворота изображения и изменение масштаба, и полный, с загрузкой всех инструментов обработки.



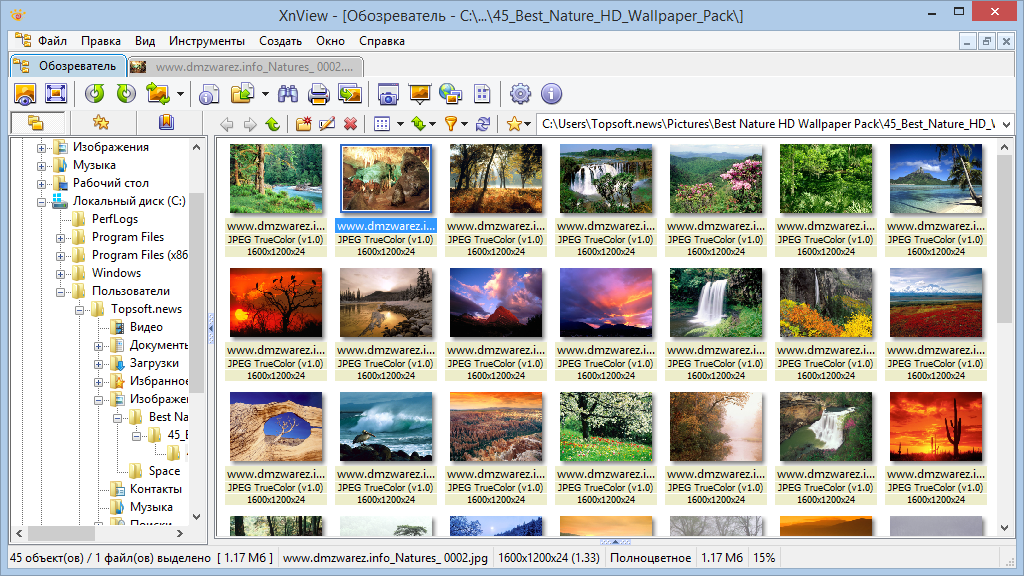
IrfanView

небольшая по размеру программа для просмотра/воспроизведения графических, видео- и аудиофайлов. Работает в среде операционных систем Windows и ReactOS, бесплатна для некоммерческого использования. Имеет набор возможностей по редактированию графических файлов.



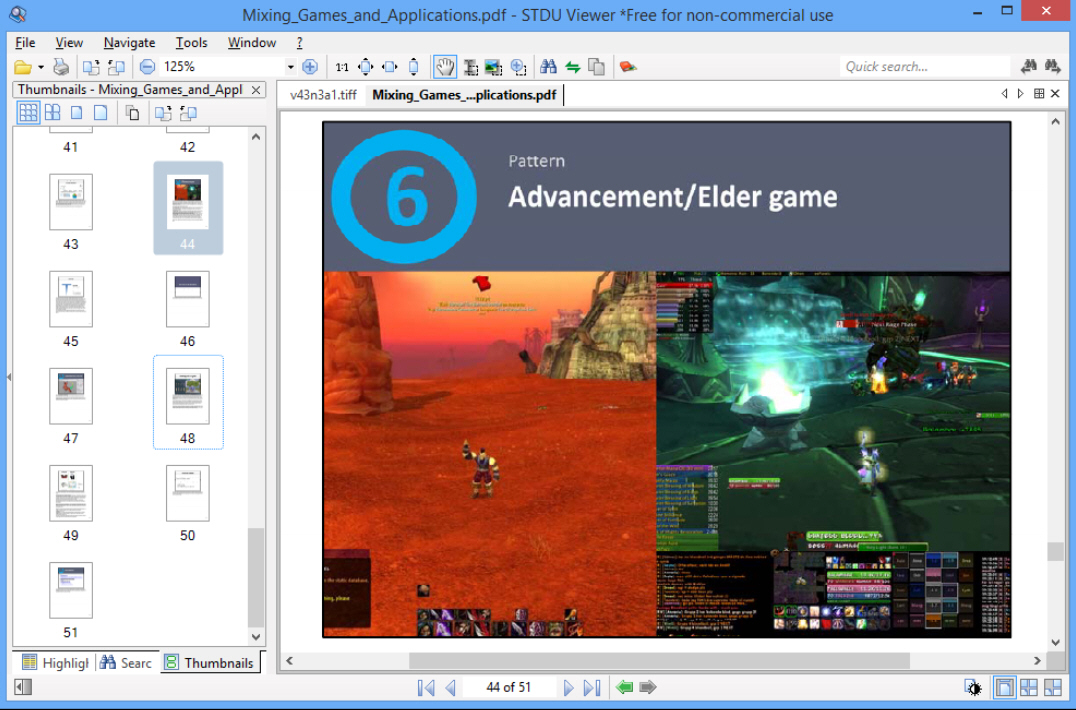
XnView

XnView — кроссплатформенная программа для просмотра изображений, поддерживающая просмотр более 400 и сохранение (конвертирование) до 50 различных графических и мультимедийных форматов файлов. Программа распространяется бесплатно для некоммерческого и образовательного использования, однако существовала и платная версия (XnView Deluxe). XnView по функциональным возможностям не уступает таким известным и популярным средствам просмотра графических форматов, как программы ACDSee и IrfanView. XnView работает со сканером и принтером, а также в полнокомплектной версии умеет проигрывать широко распространённые аудио-видео форматы файлов, но только при условии наличия нужных кодеков в системе.



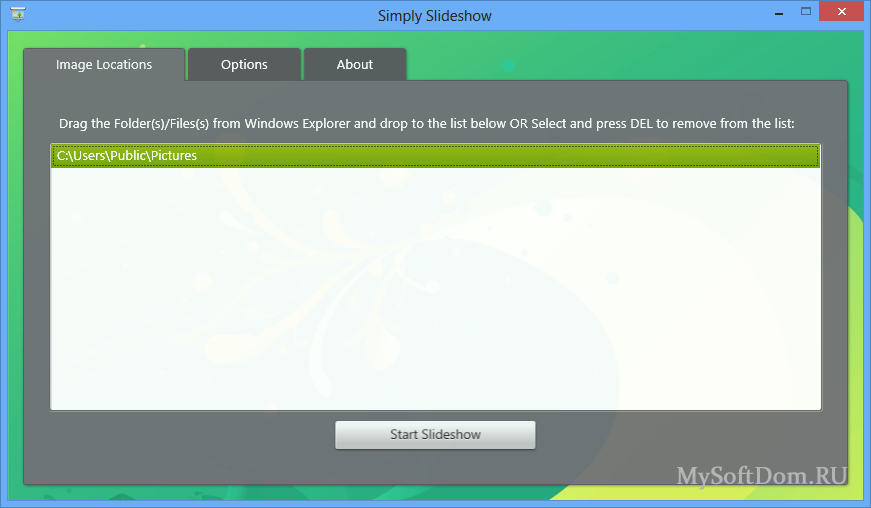
STDU Viewer

STDU Viewer — небольшой по размеру просмотрщик PDF, DjVu, Comic Book Archive (CBR или CBZ), FB2, ePub, XPS, TCR, многостраничных TIFF, TXT, PalmDoc, EMF, WMF, BMP, GIF, JPG, JPEG, PNG, PSD, PCX, DCX, MOBI, AZW для Microsoft Windows, бесплатный для некоммерческого использования. До версии 1.6 был бесплатен полностью.



Simply Slideshow

Simply Slideshow — простая утилита для запуска слайд-шоу, в том числе и в полноэкранном режиме.

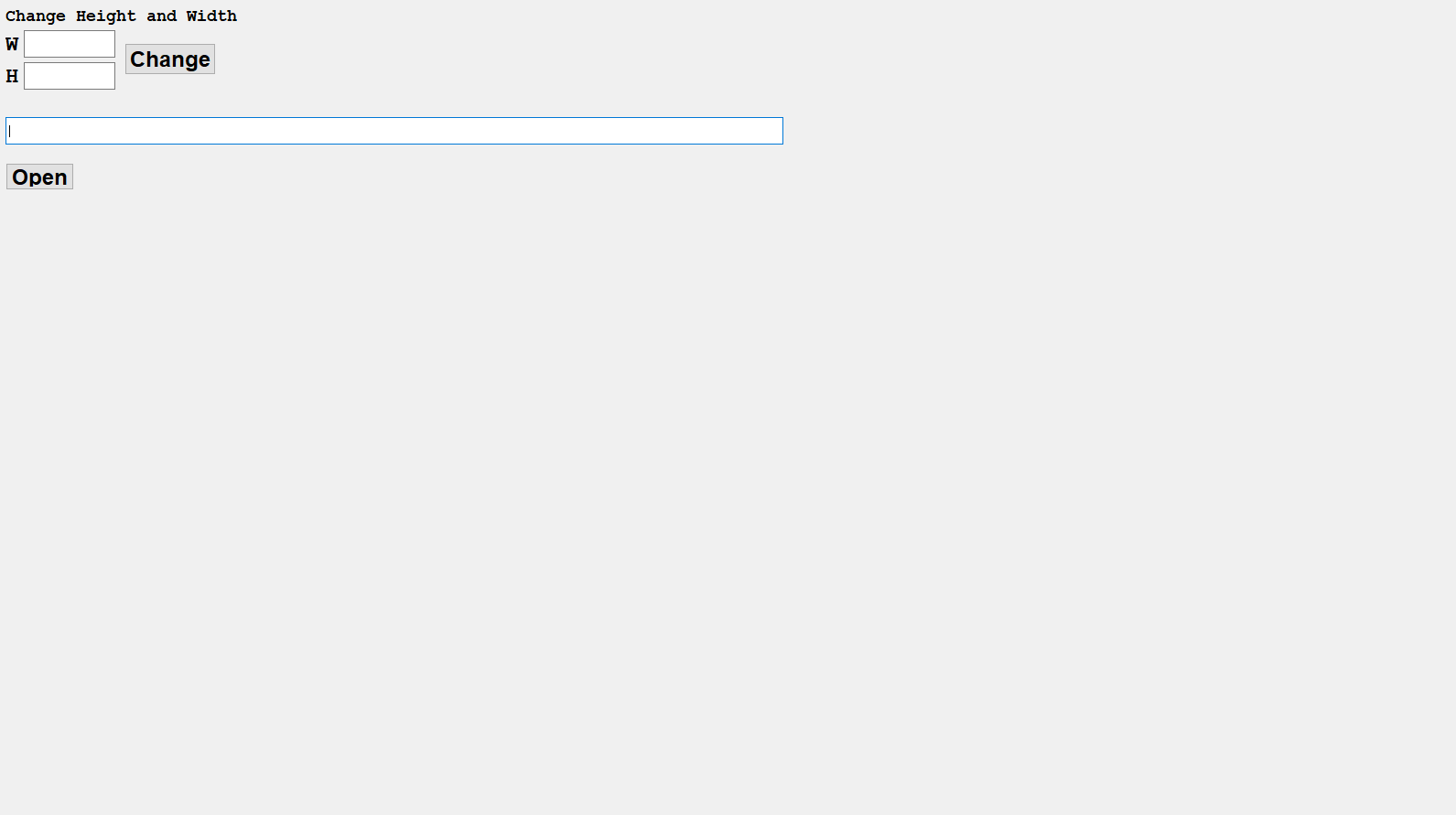


### Постановка задачи и требования, предъявляемые к программе

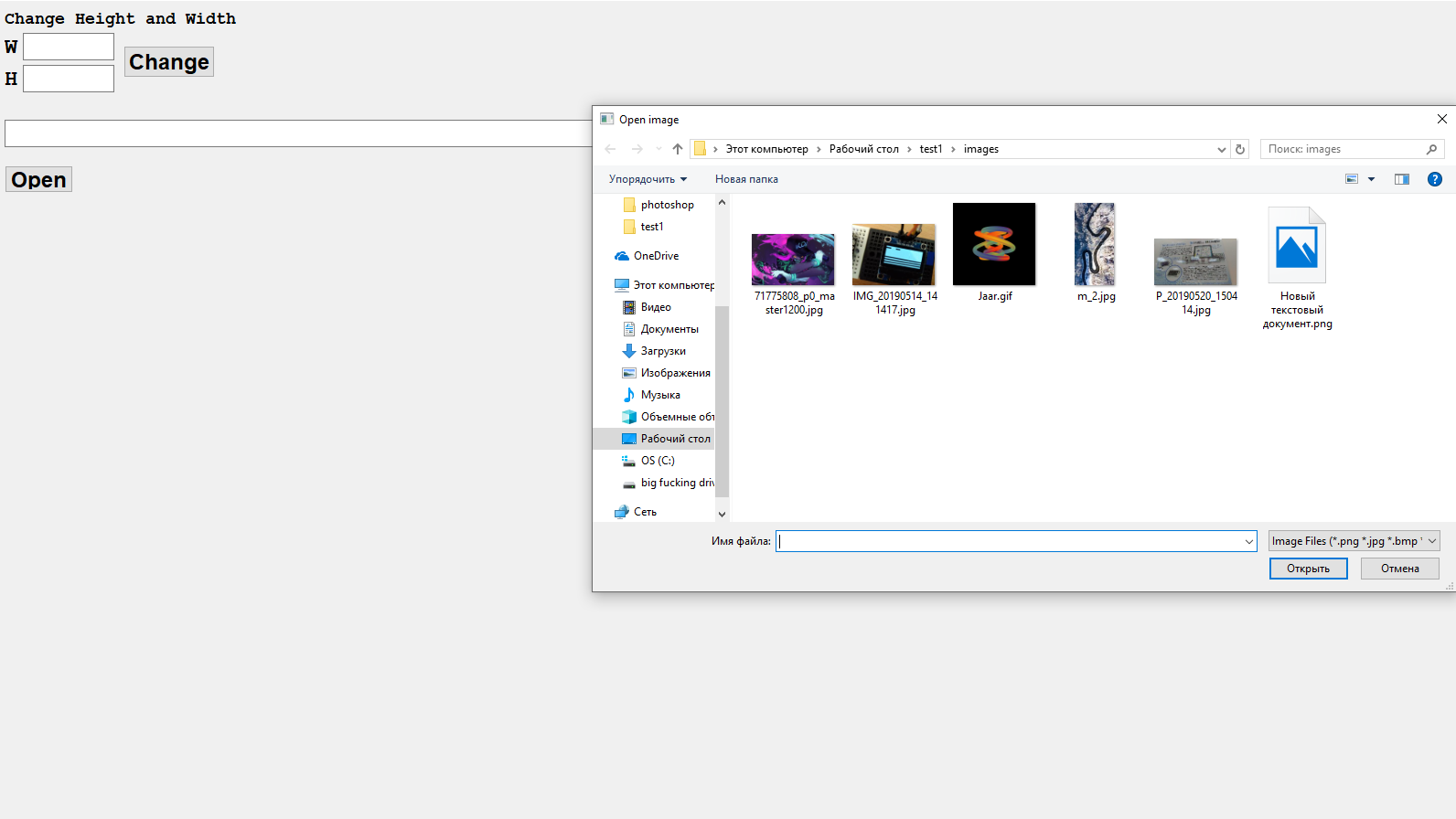
Требования:

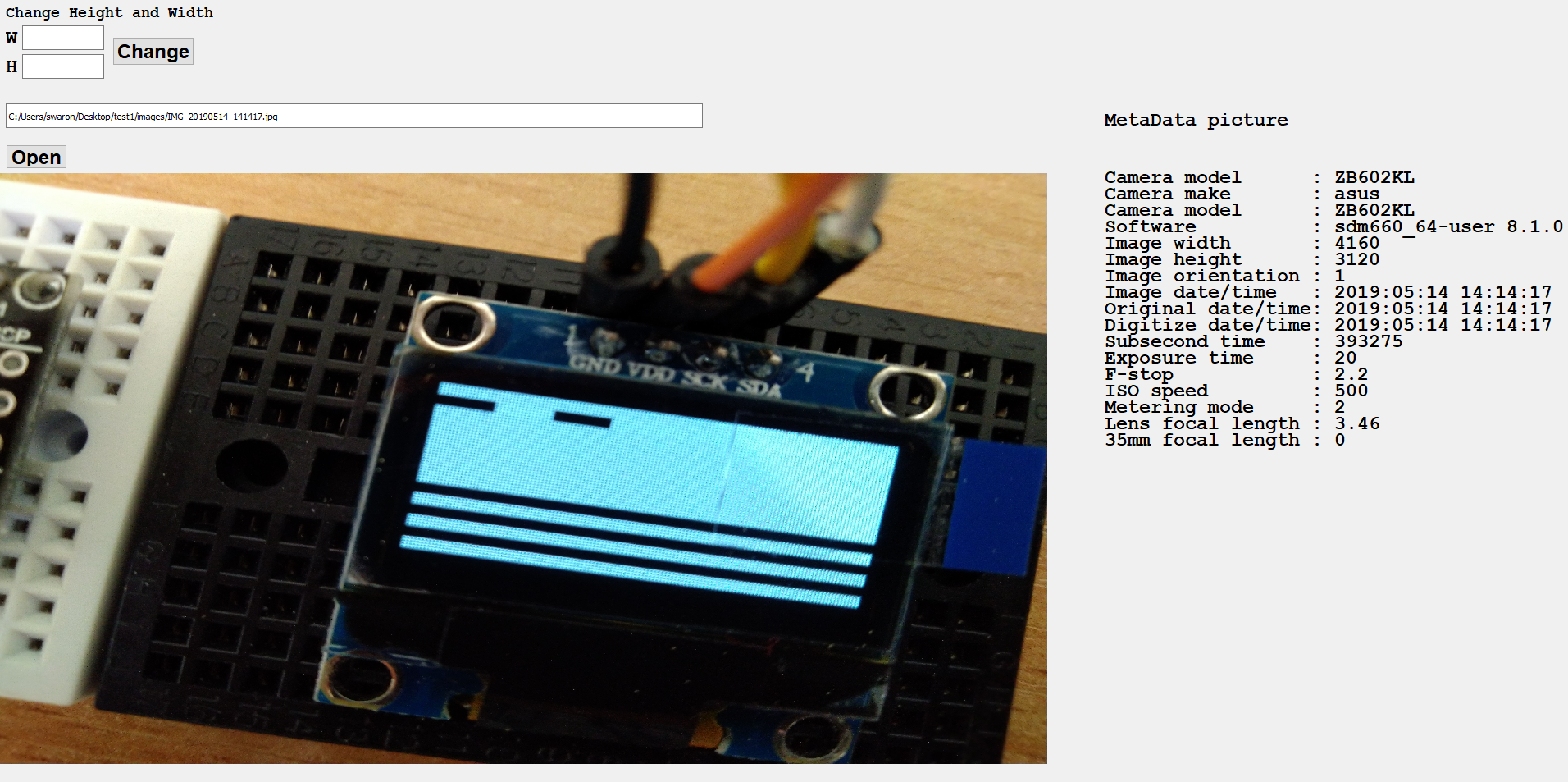
* Кнопка открытия изображения
* Диалоговое окно для выбора изображения
* Область вывода изображения
* Область вывода метаданных
* Масштабирование изображения под размер области вывода

### Описание алгоритма работы программы

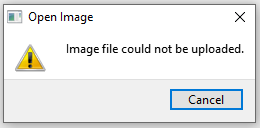
После запуска программы в открывшемся окне мы видим следущее: 

При нажатии на кнопку “Open” открывается диалогове окно где можно выбать изображение

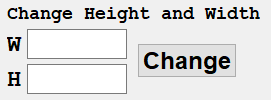


Выбираем изображение и нажимаем открыть, после видим наше изображение и его метаданные

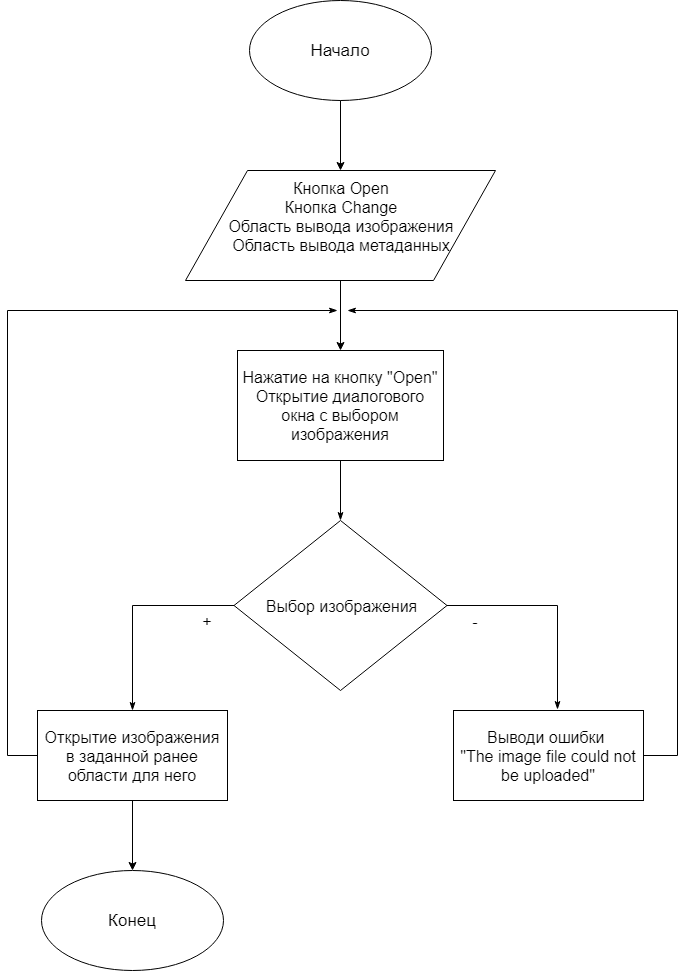
Если изображение повреждено или вовсе не является изображение а просто файл с таким же расширением то выйдет ошибка



Так же в окне мы видим такие поля как H и W и кнопку Change, где H - height (высота) а W - width (ширина). Введя в эти поля значение и нажав на кнопку “Change” мы изменим высоту и ширину изображения при загрузке



Блок схема



### Выбор и обоснование выбора средства разработки, исходный код

В качестве среды разработки я выбрал C++ с библиотекой Qt т.к. С++ наиболее распространённый язык программирования так же с библиотекой Qt в последнее время мы занимались не мало.

Код заголовочного файла mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QtWidgets>

QT\_BEGIN\_NAMESPACE

class QLabel;

class QPainter;

class QLineEdit;

class QPushButton;

QT\_END\_NAMESPACE

class MainWindow:public QMainWindow

{

Q\_OBJECT

public:

MainWindow();

private slots:

void openButtonClick();

void changeButtonClick();

protected:

void paintEvent(QPaintEvent \*event);

void openImage(const QString &path = QString());

private:

QLabel \*wLabel;

QLineEdit \*wEdit;

QLabel \*hLabel;

QLineEdit \*hEdit;

QLabel \*labLabel;

QLabel \*lab1Label;

QLabel \*fileLabel;

QLabel \*metaLabel;

QLabel \*meta1Label;

QLabel \*meta2Label;

QLabel \*meta3Label;

QLabel \*meta4Label;

QLabel \*meta5Label;

QLabel \*meta6Label;

QLabel \*meta7Label;

QLabel \*meta8Label;

QLabel \*meta9Label;

QLabel \*meta10Label;

QLabel \*meta11Label;

QLabel \*meta12Label;

QLabel \*meta13Label;

QLabel \*meta14Label;

QLabel \*meta15Label;

QLabel \*meta16Label;

QPainter \*myPainter;

QPixmap newImage;

QLineEdit \*pathEdit;

QPushButton \*openButton;

QPushButton \*changeButton;

};

#endif

Код исходника main.cpp

#include <QtWidgets>

#include "mainwindow.h"

#include "exif.h"

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

{

QApplication app(argc,argv);

MainWindow mainWin;

mainWin.show();

return app.exec();

}

Код исходника mainwindow.cpp

#include <QtWidgets>

#include "mainwindow.h"

#include "exif.h"

int argc;

int H = 720;

int W = 1280;

QString strH;

QString strW;

QString filePath;

QString strMetaData;

QString strMetaData1;

QString strMetaData2;

QString strMetaData3;

QString strMetaData4;

QString strMetaData5;

QString strMetaData6;

QString strMetaData7;

QString strMetaData8;

QString strMetaData9;

QString strMetaData10;

QString strMetaData11;

QString strMetaData12;

QString strMetaData13;

QString strMetaData14;

QString strMetaData15;

QString strMetaData16;

MainWindow::MainWindow()

{

setWindowTitle(tr("Gallery"));

resize(1600,900);

fileLabel=new QLabel(this);

fileLabel->setGeometry(10,5,850,120);

fileLabel->setFont(QFont("Courier",18,QFont::Bold));

fileLabel->setText("Open: "+filePath);

pathEdit=new QLineEdit(this);

pathEdit->setText(filePath);

pathEdit->setGeometry(10,130,850,30);

fileLabel->setFont(QFont("System",12,QFont::Bold));

fileLabel->setText(filePath);

labLabel = new QLabel(this);

labLabel->setGeometry(10, 0, 850, 40);

labLabel->setFont(QFont("Courier", 14, QFont::Bold));

labLabel->setText("Change Height and Width");

lab1Label = new QLabel(this);

lab1Label->setGeometry(1350, 130, 850, 40);

lab1Label->setFont(QFont("Courier", 18, QFont::Bold));

hLabel=new QLabel(this);

hLabel->setGeometry(10,65,60,40);

hLabel->setFont(QFont("Courier",18,QFont::Bold));

hLabel->setText("H");

hEdit=new QLineEdit(this);

hEdit->setGeometry(30,70,100,30);

hEdit->setFont(QFont("System",12,QFont::Normal));

hEdit->setText(strH);

wLabel=new QLabel(this);

wLabel->setGeometry(10,30,60,40);

wLabel->setFont(QFont("Courier",18,QFont::Bold));

wLabel->setText("W");

wEdit=new QLineEdit(this);

wEdit->setGeometry(30,35,100,30);

wEdit->setFont(QFont("System",12,QFont::Normal));

wEdit->setText(strW);

metaLabel = new QLabel(this);

metaLabel->setGeometry(1350, 160, 850, 120);

metaLabel->setFont(QFont("Courier", 18, QFont::Bold));

meta1Label = new QLabel(this);

meta1Label->setGeometry(1350, 180, 850, 120);

meta1Label->setFont(QFont("Courier", 18, QFont::Bold));

meta2Label = new QLabel(this);

meta2Label->setGeometry(1350, 200, 850, 120);

meta2Label->setFont(QFont("Courier", 18, QFont::Bold));

meta3Label = new QLabel(this);

meta3Label->setGeometry(1350, 220, 850, 120);

meta3Label->setFont(QFont("Courier", 18, QFont::Bold));

meta4Label = new QLabel(this);

meta4Label->setGeometry(1350, 240, 850, 120);

meta4Label->setFont(QFont("Courier", 18, QFont::Bold));

meta5Label = new QLabel(this);

meta5Label->setGeometry(1350, 260, 850, 120);

meta5Label->setFont(QFont("Courier", 18, QFont::Bold));

meta6Label = new QLabel(this);

meta6Label->setGeometry(1350, 280, 850, 120);

meta6Label->setFont(QFont("Courier", 18, QFont::Bold));

meta7Label = new QLabel(this);

meta7Label->setGeometry(1350, 300, 850, 120);

meta7Label->setFont(QFont("Courier", 18, QFont::Bold));

meta8Label = new QLabel(this);

meta8Label->setGeometry(1350, 320, 850, 120);

meta8Label->setFont(QFont("Courier", 18, QFont::Bold));

meta9Label = new QLabel(this);

meta9Label->setGeometry(1350, 340, 850, 120);

meta9Label->setFont(QFont("Courier", 18, QFont::Bold));

meta10Label = new QLabel(this);

meta10Label->setGeometry(1350,360, 850, 120);

meta10Label->setFont(QFont("Courier", 18, QFont::Bold));

meta11Label = new QLabel(this);

meta11Label->setGeometry(1350, 380, 850, 120);

meta11Label->setFont(QFont("Courier", 18, QFont::Bold));

meta12Label = new QLabel(this);

meta12Label->setGeometry(1350, 400, 850, 120);

meta12Label->setFont(QFont("Courier", 18, QFont::Bold));

meta13Label = new QLabel(this);

meta13Label->setGeometry(1350, 420, 850, 120);

meta13Label->setFont(QFont("Courier", 18, QFont::Bold));

meta14Label = new QLabel(this);

meta14Label->setGeometry(1350, 440, 850, 120);

meta14Label->setFont(QFont("Courier", 18, QFont::Bold));

meta15Label = new QLabel(this);

meta15Label->setGeometry(1350, 460, 850, 120);

meta15Label->setFont(QFont("Courier", 18, QFont::Bold));

meta16Label = new QLabel(this);

meta16Label->setGeometry(1350, 480, 850, 120);

meta16Label->setFont(QFont("Courier", 18, QFont::Bold));

QPushButton\* changeButton = new QPushButton(tr("Change"), this);

changeButton->setGeometry(140, 49, 100, 35);

changeButton->setFont(QFont("Times", 18, QFont::Bold));

connect(changeButton, SIGNAL(clicked()), this, SLOT(changeButtonClick()));

QPushButton \*openButton=new QPushButton(tr("Open"),this);

openButton->setGeometry(10,180,75,30);

openButton->setFont(QFont("Times",18,QFont::Bold));

connect(openButton,SIGNAL(clicked()),this,SLOT(openButtonClick()));

}

void MainWindow::changeButtonClick()

{

strH = hEdit->text();

H = strH.toInt();

strW = wEdit->text();

W = strW.toInt();

}

void MainWindow::openButtonClick()

{

filePath = pathEdit->text();

openImage();

pathEdit->setText(filePath);

update();

QFile file(filePath);

if (file.open(QIODevice::ReadOnly)) {

QByteArray data = file.readAll();

easyexif::EXIFInfo info;

if (int code = info.parseFrom((unsigned char\*)data.data(), data.size()))

{

}

strMetaData = info.Model.c\_str();

strMetaData1 = info.Make.c\_str();

strMetaData2 = info.Model.c\_str();

strMetaData3 = info.Software.c\_str();

strMetaData4.setNum(info.ImageWidth);

strMetaData5.setNum(info.ImageHeight);

strMetaData6.setNum(info.Orientation);

strMetaData7 = info.DateTime.c\_str();

strMetaData8 = info.DateTimeOriginal.c\_str();

strMetaData9 = info.DateTimeDigitized.c\_str();

strMetaData10 = info.SubSecTimeOriginal.c\_str();

strMetaData11.setNum((unsigned)(1.0 / info.ExposureTime));

strMetaData12.setNum(info.FNumber);

strMetaData13.setNum(info.ISOSpeedRatings);

strMetaData14.setNum(info.MeteringMode);

strMetaData15.setNum(info.FocalLength);

strMetaData16.setNum(info.FocalLengthIn35mm);

lab1Label->setText("MetaData picture");

metaLabel->setText("Camera model : " + strMetaData);

meta1Label->setText("Camera make : " + strMetaData1);

meta2Label->setText("Camera model : " + strMetaData2);

meta3Label->setText("Software : " + strMetaData3);

meta4Label->setText("Image width : " + strMetaData4);

meta5Label->setText("Image height : " + strMetaData5);

meta6Label->setText("Image orientation : " + strMetaData6);

meta7Label->setText("Image date/time : " + strMetaData7);

meta8Label->setText("Original date/time: " + strMetaData8);

meta9Label->setText("Digitize date/time: " + strMetaData9);

meta10Label->setText("Subsecond time : " + strMetaData10);

meta11Label->setText("Exposure time : " + strMetaData11);

meta12Label->setText("F-stop : " + strMetaData12);

meta13Label->setText("ISO speed : " + strMetaData13);

meta14Label->setText("Metering mode : " + strMetaData14);

meta15Label->setText("Lens focal length : " + strMetaData15);

meta16Label->setText("35mm focal length : " + strMetaData16);

}

}

void MainWindow::openImage(const QString &path)

{

filePath=path;

if (filePath.isNull())

{filePath=QFileDialog::getOpenFileName(this,tr("Open image"),"","Image Files (\*.png \*.jpg \*.bmp)");

if (!filePath.isEmpty())

{ if(!newImage.load(filePath))

{QMessageBox::warning(this,tr("Open Image"),tr("Image file could not be uploaded."),QMessageBox::Cancel);

return;

}

}

}

}

void MainWindow::paintEvent(QPaintEvent \*)

{

QPixmap bgPixmap(filePath);

QPixmap scaled = bgPixmap.scaled(QSize(W, H));

QPainter myPainter(this);

myPainter.drawPixmap(0,215,scaled);

}

### Тестирование программы

В ходе тестирования программы был выявлен ряд недочетов:

* При открытии изображения формата .png не выводятся метаданные
* Нет автоматического масштабирования изображения
* При открытии файлов формата .gif изображение не воспроизводится

### Руководство по установке

Перед тем как начать пользоваться программой надо скачать её по ссылку “https://github.com/stfuhentai/-2019” в виде архива. После скачивания архива его следует разархивировать в любое удобное для вас место. После извлечения папки запустите “.exe” файл который находиться в папе “release” и готово.

### Руководство пользователя

После запуска программы для открытия изображения нужно нажать кнопку “Open” в открывшемся диалоговом окне выбрать нужное нам изображения.  
В случае если разрешения изображения не устраивает его можно изменить с помощью двух полей H и W в верхнем левом углу. Метаданные выводятся правее от изображения.

### Ссылки на интернет-ресурсы

1. “EXIF библиотека” GitHub.com

URL: https://github.com/mayanklahiri/easyexif

1. “Чтение метаданных с помощью EXIF” geekquestion.com URL: https://geekquestion.com/10387901-chitajte-exif-metadannye-izobrazhenij-v-qt/
2. “Qt” qt.io URL: https://www.qt.io
3. “Масштабирование изображения” web-answers.ru URL: https://web-answers.ru/c/pochemu-qpixmap-scaled-ne-rabotaet.html
4. “QImageReader Class” doc.qt.io URL: https://doc.qt.io/qt-5/qimagereader.html#setAutoTransform
5. “Image Viewer Example” doc.qt.io URL: https://doc.qt.io/qt-5/qtwidgets-widgets-imageviewer-example.html