Министерство науки и высшего образования 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(а): |
| студент(ка) ОП СПО группа 33 |
| Кузнецов К.В |
|  |
| Руководитель практики: старший преподаватель кафедры математики и информатики Касаткин К.А. |

г. Глазов 2020

**Оглавление**

1. [Техническое задание](#_heading=h.4f00mqeva40w)
2. [Руководство пользователя](#_heading=h.ldolnwpbmy5y)
3. [Руководство системного администратора](#_heading=h.n0xzo3t760wx)
4. [Аналоги ПО](#_heading=h.zbiw4576z44z)
5. [Описание программы](#_heading=h.o0c7oq4xnale)
6. [Формализация бизнес-процессов предметной области](#_heading=h.e39hub91hafq)
7. [Исходный код](#_heading=h.f2841d5l5doe)
8. [Отчет по тестированию](#_heading=h.y8qyo4laa7yq)
9. [Список источников](#_heading=h.40bxc8kwud6a)

### Техническое задание на создание программного обеспечения на тему «Фотогалерея»

1. **Цель : создание программы для просмотра фотографий**
2. **Основные задачи :**
   1. Выбор каталога в котором находятся фотографии
   2. Выбор фотографии
   3. Вывод фотографии в окно приложения
3. **Используемые технические и программные средства:**
   1. Персональный компьютер стандартной конфигурации
   2. Операционная система Windows 10 v.1909
   3. Язык программирования C++
   4. Среда разработки Qt Creator
4. **Требования к системе:**
   1. Минимальные требования к техническим средствам
   2. Простой интерфейс
   3. Вывод информации о размере и дате создания изображения
5. **Состав документации:**
   1. Руководство пользователя
   2. Руководство системного администратора

### Руководство пользователя

1. Откройте приложение кликнув по иконке.
2. Нажмите на кнопку “Открыть каталог”
   1. Выберите папку в которой находятся изображения
   2. Нажмите на соответствующую кнопку “Выбор папки”
3. Выберите изображение, которое хотите отобразить в окне программы
   1. Кликните на изображение находящееся в списке в левой части окна программы
   2. Снизу отобразится информация по выбранному изображению.

# Руководство системного администратора

1. Скачайте и установите программный продукт «Фотогалерея»
2. Создайте ярлык на рабочем столе.
3. Убедитесь в правильности запуска программного продукта.

#### Аналоги ПО

**FastStone Image Viewer**  
FastStone Image Viewer — программа для просмотра изображений в Microsoft Windows. Включает в себя файловый менеджер и базу данных. Считается одной из наиболее функциональных бесплатных программ в своём классе.

**Преимущества:**

1. Скорость работы
2. Бесплатность
3. Нетребовательная к ресурсам ПК
4. Интуитивно понятная
5. Широкая функциональность
6. Стабильность

**Недостатки:**

1. Нет возможности приплюсовать фото
2. Некоторые настройки программы не столь разнообразны
3. При написании текста, нельзя изменить параметры по ходу действия. Приходиться убирать текст полностью.

**Phototheca**  
Phototheca - это органайзер изображений и средство просмотра изображений для организации цифровых фотографий. Он работает на платформе Microsoft Windows. Программное обеспечение изначально было создано компанией Lunarship Software и впервые выпущено в 2012 году.

**Преимущества:**

1. Бесплатность
2. Удобный интерфейс
3. Широкая функциональность
4. Скорость работы

**Недостатки:**

1. Не поддерживает экраны высокого разрешения

### Описание программы

Программа «Фотогалерея» предназначена для просмотра фотографий.  
Чтобы просмотреть фотографии, достаточно указать путь к каталогу.  
После выбора изображения внизу программы отобразится следующая информация - имя изображения, его формат, размер, дата создания выбранного изображения.

### Формализация бизнес-процессов предметной области

### Исходный код

**#photogallery1.pro**

**QT += core gui**

**greaterThan(QT\_MAJOR\_VERSION, 4): QT += widgets**

**CONFIG += c++11**

**DEFINES += QT\_DEPRECATED\_WARNINGS**

**SOURCES += \**

**main.cpp \**

**window.cpp**

**HEADERS += \**

**window.h**

**qnx: target.path = /tmp/$${TARGET}/bin**

**else: unix:!android: target.path = /opt/$${TARGET}/bin**

**!isEmpty(target.path): INSTALLS += target**

**//window.h**

**#include <QtWidgets>**

**class Window : public QWidget{**

**Q\_OBJECT**

**public:**

**Window();**

**private:**

**QHBoxLayout \*hbox;**

**QLabel \*labelImage;**

**QPixmap pixmap;**

**QTreeWidget \*treeWidgetListImage;**

**QLineEdit \*editPath;**

**QString filePath;**

**QLabel \*labelInfo;**

**private slots:**

**void viewListImage();**

**void clickOnItemImage();**

**void resizeEvent(QResizeEvent \*event);**

**void updateImage();**

**void clickButtonOpen();**

**};**

**//window.cpp**

**#include "window.h"**

**Window::Window(){**

**hbox = new QHBoxLayout();**

**QHBoxLayout \*hbox2 = new QHBoxLayout();**

**QPushButton \*button = new QPushButton("Открыть каталог");**

**button->connect(button, &QPushButton::clicked, this, &Window::clickButtonOpen);**

**editPath = new QLineEdit();**

**editPath->setReadOnly(true);**

**QLabel \*label = new QLabel("Текущий каталог");**

**hbox2->addWidget(label);**

**hbox2->addWidget(editPath);**

**hbox2->addWidget(button);**

**treeWidgetListImage = new QTreeWidget();**

**treeWidgetListImage->setColumnCount(1);**

**treeWidgetListImage->setHeaderLabel("Изображения");**

**treeWidgetListImage->connect(treeWidgetListImage, &QTreeWidget::itemClicked, this, &Window::clickOnItemImage);**

**treeWidgetListImage->setStyleSheet("background-color: #f0f0f0; border: 0px solid black;");**

**labelImage = new QLabel();**

**labelImage->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Expanding);**

**labelImage->setMinimumSize(240, 160);**

**labelImage->setAlignment(Qt::AlignCenter);**

**QFrame \*line = new QFrame();**

**line->setFrameShape(QFrame::VLine);**

**line->setFrameShadow(QFrame::Sunken);**

**labelInfo = new QLabel();**

**hbox->addWidget(treeWidgetListImage, 1);**

**hbox->addWidget(line);**

**hbox->addWidget(labelImage, 3);**

**QVBoxLayout \*vbox1 = new QVBoxLayout(this);**

**vbox1->addLayout(hbox2);**

**vbox1->addLayout(hbox);**

**vbox1->addWidget(labelInfo);**

**}**

**void Window::viewListImage()**

**{**

**QDir dir(filePath);**

**QStringList listSuffix = {"\*.jpg", "\*.png", "\*.png"};**

**dir.setNameFilters(listSuffix);**

**dir.setFilter(QDir::Files | QDir::Hidden | QDir::NoSymLinks);**

**QFileInfoList list = dir.entryInfoList();**

**treeWidgetListImage->clear();**

**for (int i = 0; i < list.size(); i++){**

**QFileInfo info = list.at(i);**

**qDebug() << info.filePath() << endl;**

**QTreeWidgetItem \*item = new QTreeWidgetItem(treeWidgetListImage);**

**item->setText(0, info.fileName());**

**}**

**}**

**void Window::clickOnItemImage()**

**{**

**QString pathImage = filePath + "/" + treeWidgetListImage->currentItem()->data(0,0).toString();**

**pixmap.load(pathImage);**

**QFileInfo info(pathImage);**

**labelInfo->setText("Filename: " + info.fileName() + "\n" + "Size: " + QString::number(info.size()) + " byte" + "\n" + "Date creation: " + info.birthTime().toString());**

**qDebug() << filePath + "/" + treeWidgetListImage->currentItem()->data(0,0).toString() << endl;**

**updateImage();**

**}**

**void Window::resizeEvent(QResizeEvent \*event){**

**QSize scaledSize = pixmap.size(); //Размер изображения**

**scaledSize.scale(labelImage->size(), Qt::KeepAspectRatio); //Меняем размер до размеров QLabel**

**if (!labelImage->pixmap() || scaledSize != labelImage->pixmap()->size()){**

**updateImage();**

**}**

**}**

**void Window::updateImage()**

**{**

**qDebug() << rand() % 10 << endl;**

**if (!pixmap.isNull()){**

**labelImage->setPixmap(pixmap.scaled(labelImage->size(), Qt::KeepAspectRatio, Qt::SmoothTransformation));**

**}**

**}**

**void Window::clickButtonOpen()**

**{**

**filePath = QFileDialog::getExistingDirectory(this, tr("Open directory"), "",**

**QFileDialog :: ShowDirsOnly | QFileDialog :: DontResolveSymlinks);**

**if (!filePath.isEmpty()){**

**editPath->setText(filePath);**

**viewListImage();**

**}**

**}**

**//main.cpp**

**#include "window.h"**

**#include <QApplication>**

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

**{**

**QApplication a(*argc*, argv);**

**Window w;**

**w.show();**

**return a.exec();**

**}**

### 

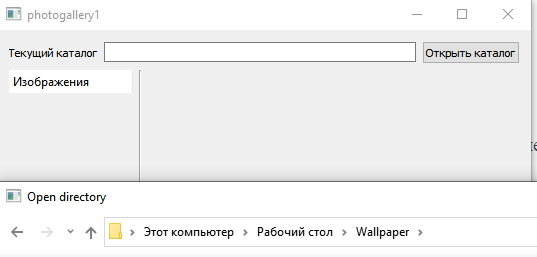
### 

### 

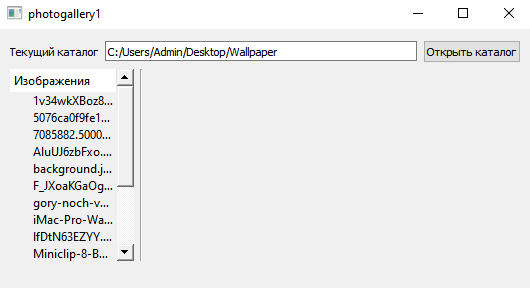
### 

### Отчет по тестированию

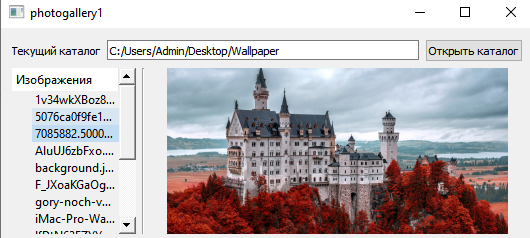
1. При тестировании был проверен выбор каталога. Тестирование прошло успешно.



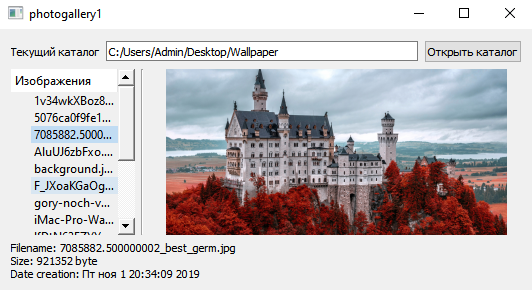
1. Были проведены проверки по получению списка изображений находящихся в выбранном каталоге. Тесты прошли успешно.



1. Проверка на вывод выбранного изображения в окно приложения. Тесты прошли успешно.



1. Проверка на вывод информации по выбранному изображению внизу окна приложения. Тесты прошли успешно.



### Список источников

1. Google Drive. **URL :** [**https://drive.google.com/drive**](https://drive.google.com/drive)
2. Проект на GitHub. **URL :** [**https://github.com/Syqcle/photogallery**](https://github.com/Syqcle/photogallery)
3. Trello. **URL :** [**https://trello.com/**](https://trello.com/)
4. Qt Documentation. **URL :** [**https://doc.qt.io/**](https://doc.qt.io/)
5. FastStone Image Viewer. **URL :** [https://www.faststone.org/**https://www.faststone.org/**](https://www.faststone.org/)
6. Phototheca. **URL :** [**https://lunarship.com/**](https://lunarship.com/)