Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Отчет защищен

с оценкой *\_\_\_\_\_\_\_\_*

Доцент, канд. ф-м. наук

\_\_\_\_\_\_\_\_ А. Б. Раухваргер

« » 2022

**ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ГРУПП РИСКА ПОСРЕДСТВОМ ЦВЕТОВОГО АНАЛИЗА ЦИФРОВЫХ ФОТОГРАФИЙ КОЖНЫХ ОБРАЗОВАНИЙ**

Пояснительная записка к отчету

о научно-исследовательской работе

ЯГТУ 09.04.02 - 006 НИР

Нормоконтролер, Отчет выполнил

доцент, канд. ф-м. наук студент гр. ЭМИС-24м

\_\_\_\_\_\_\_\_\_\_ А. Б. Раухваргер \_\_\_\_\_\_\_\_\_\_ В. А. Матвеев

« » 2022 « » 2022

2022

Содержание

[1 Введение 3](#_Toc104149414)

[1.1 Характеристика объекта исследования 4](#_Toc104149415)

[1.2 Стек используемых технологий 5](#_Toc104149416)

[1.2.1 Плюсы и минусы языка Java 5](#_Toc104149417)

[1.2.2 Плюсы и минусы языка C# 5](#_Toc104149418)

[1.2.3 Плюсы и минусы языка Python 6](#_Toc104149419)

[1.3 Анализ существующих решений 7](#_Toc104149420)

[1.3.1 Платформа «Про родинки» 7](#_Toc104149421)

[1.3.2 Платформа «Celsus» 9](#_Toc104149422)

[1.4 Постановка цели и задачи 10](#_Toc104149423)

[1.5 Вывод по аналитической части 10](#_Toc104149424)

[2 Проектная часть 12](#_Toc104149425)

[2.1 Описание предметной области 12](#_Toc104149426)

[2.2 Концептуальное проектирование 12](#_Toc104149427)

[2.3 Проектирование базы данных 12](#_Toc104149428)

[2.4 Выбор системы управлению базой данных 12](#_Toc104149429)

[2.5 Физическая модель базы данных 12](#_Toc104149430)

[2.6 Вывод по проектной части 12](#_Toc104149431)

[3 Технологическая часть 13](#_Toc104149432)

[3.1 Взаимодействие базы данных и программы 13](#_Toc104149433)

[3.2 Используемые классы объектов 15](#_Toc104149434)

[3.3 Описание работы программы 18](#_Toc104149435)

[3.4 Вывод по технологической части 22](#_Toc104149436)

[4 Исследовательская часть 23](#_Toc104149437)

[4.1 Входные данные 23](#_Toc104149438)

[4.2 Данные и инструменты анализа 23](#_Toc104149439)

[4.3 Вывод по технологической части 23](#_Toc104149440)

[Список используемой литературы 24](#_Toc104149441)

# Введение

Выбранная тема «Исследование возможности автоматического распознавания групп риска посредством цветового анализа цифровых фотографий кожных образований» актуальна в наше время, так как важно обнаружить и диагностировать онкологию на ранних стадиях.

По ходу обработки используется алгоритм приведения изображения к средней яркости. Это необходимо для исследования всех экземпляров заболеваний в равных условиях, на которые не будет влиять освещенность, на момент создания фотографии.

Проводя обработку изображения, на гистограммах яркости (0-255) можно экспертным способом наблюдать какой из параметров частоты цветовой модели R, G, B возрастает или убывает, а также где именно это происходит. Из дополнительных, доступных данных для исследования, следует обращать внимание на значения средней яркости, среднеквадратичное отклонение и медиану каждой шкалы R, G, B.

Предполагается, что, владея данными, описанными ранее, можно будет определить предположительное заболевание онкологического характера.

1. Аналитическая часть

## Характеристика объекта исследования

Главный объект исследования представляет собой экземпляры цифровых фотографий, которые необходимо проанализировать для установления возможных закономерностей на основе цветовой модели RGB.

Необходимость в диагностике онкологических заболеваний с помощью автоматизированных средств подтвердили между собой кафедра информационных технологий, онкологическая больница и медицинский институт.

На данный момент, предполагается исследование для следующих групп онкологических заболеваний:

* Базальноклеточный рак – это наиболее часто встречающаяся y человека раковая опухоль. Состоит из клеток, подобных клеткам базального слоя эпидермиса. От других раков кожи отличается чрезвычайно редким метастазированием, однако способна к обширному местному росту, который приводит к существенным косметическим и функциональным нарушениям.
* Меланома – это злокачественная опухоль, развивающаяся из меланоцитов - пигментных клеток, продуцирующих меланины. Наряду с плоскоклеточным и базальноклеточным раком кожи относится к злокачественным опухолям кожи. Преимущественно локализуется в коже, реже - сетчатке глаза, слизистых оболочках. Одна из наиболее опасных злокачественных опухолей человека, часто рецидивирующая и метастазирующая лимфогенным и гематогенным путём почти во все органы.

По ходу исследования необходимо разработать программу, реализующую функционал для нахождения зависимостей между цветовой модели RGB и онкологическими заболеваниями на изображениях.

К задачам исследования необходимо отнести следующее:

* Привести все экземпляры изображений к средней яркости;
* создать графические элементы в виде графиков для вывода информации для исследования;
* создать базу данных для хранения информации о соотношении значений параметров RGB и экземпляра изображения;
* сохранять и загружать результаты исследования.

## Стек используемых технологий

Для реализации исследования будут рассмотрены несколько языков программирования. Основными критериями будут выступать задачи исследования, описанные выше. Языками программирования, среди которых было необходимо выбирать, были следующие: Java, C#, Python.

### Плюсы и минусы языка Java

Java – это язык программирования общего назначения, объектно-ориентированный, является актуальным спустя 20 лет. Используется везде: мобильные, сетевые, настольные и корпоративные приложения.

Язык Java начал свой путь в начале 1990-х, компания Sun Microsystems начала разработку улучшенной версии C++, то есть независимую от одной платформы, автоматическим управлением памятью и достаточно простым в освоении.

Плюсы языка Java:

* Java включает в себя объектно-ориентированное программирование;
* Java – язык высокого уровня с простым синтаксисом;
* отсутствие указателей, что означает повышенную безопасность;
* Java дешевый язык и работать с ним можно с любого компьютера, он поддерживается в большом числе компаний;
* Java является кроссплатформенным языком;
* встроенная дистрибуция;
* автоматическое управление памятью;
* многопоточность;
* встроенная поддержка работы в сетях;
* мощные стандартные библиотеки;
* поддержка Oracle и большое сообщество.

Минусы языка Java:

* Низкая производительность;
* платное коммерческое использование;
* отсутствие нативного дизайна;
* повышенные требования к объему оперативной памяти;

### Плюсы и минусы языка C#

Язык программирования C# в наше время является актуальным и популярным. Так как C# принадлежит компании Microsoft, то он применяется почти во всех продуктах, которые были разработаны или куплены.

Основные положительные стороны языка программирования C#:

* Как и было сказано выше, C# поддерживается и успешно развивается Microsoft и по сегодняшний день;
* типы данных имеют фиксированный размер, что повышает мобильность;
* автоматическая очистка памяти
* синтаксис похож на C, C++ или Java, поэтому C# является перспективным языком для изучения;
* среда программирования Visual Studio, на которой, в основном, используют C#, распространяется бесплатно и с открытым исходным кодом для небольших компаний и частных лиц;
* благодаря приобретению Xamarin, на C# можно писать код под Android и iOS;
* большое сообщество программистов по всему миру;
* наличие синтаксического сахара;
* подсказки и подробное описание каждой функции;
* наличие объектно-ориентированного программирования;
* большое количество вакантных мест на должность C#-программиста.

Минусы языка программирования C#:

* Главная ориентированность - .Net (платформа Windows);
* сохранился оператор безусловного перехода go to.

Бесплатно распространяется только на частные лица, небольшие компании и учащихся.

### Плюсы и минусы языка Python

Python – высокоуровневый объектно-ориентированный язык общего назначения, который используется в том числе и для разработки веб-приложений. Язык ориентирован на повышение производительности разработчика и читаемости кода.

Плюсы языка Python:

* Низкий порог вхождения;
* простой синтаксис;
* кроссплатформенный язык;
* большое сообщество и поддержка;
* широкое применение;
* востребованность на рынке труда;
* интеграция с другими языками;
* большое число библиотек.

Минусы языка Python:

* Низкая скорость выполнения программ;
* проблема работы с русскими символами;
* необходимость сторонних приложений для конвертации .py в .exe;
* при копировании кода не учитывается табуляция;
* большое потребление памяти;
* для разработки мобильных приложений нужна сторонняя программа;
* ограничения по дизайну.

## Анализ существующих решений

По результатам поиска в интернете, были найдены следующие программные аналоги [4,5]:

* «Про родинки»;
* «Celsus».

### Платформа «Про родинки»

Методика получения рекомендаций платформы "Про родинки" была предложена экспертами дерматологами и онкологами "Приволжского исследовательского медицинского университета" Министерства здравоохранения Российской Федерации. Разработка велась в тесном сотрудничестве с врачами-экспертами. В процессе эксплуатации врачи осуществляют непрерывный контроль за работой платформы.

Основой платформы является нейросеть, специально построенная и обученная распознаванию рисков злокачественных заболеваний кожи по фотографиям подозрительных "родинок", сделанных смартфоном.

Нейросеть была обучена на более чем 4000 случаев с подтвержденными диагнозами.

Направлениями обучения было распознавание меланомы и базально-клеточного рака на ранних стадиях.

Работа платформы "Про родинки" была протестирована в ходе испытаний на 1000 реальных случаев.

В результате испытаний были получены следующие характеристики качества работы платформы "Про родинки":

Чувствительность - вероятность правильного заключения нейросети при предъявлении злокачественного заболевания (базально-клеточного рака или меланомы) составила 88%. Соответственно вероятность ложно отрицательной ошибки - 12%

Специфичность - вероятность правильного заключения нейросети при отсутствии злокачественного заболевания (базально-клеточного рака или меланомы) составила 78%. Соответственно вероятность ложно положительной ошибки - 22%.

Результат испытаний подтвержден специальным Отчетом Федерального государственного бюджетного образовательного учреждения высшего образования "Приволжский исследовательский медицинский университет" Министерства здравоохранения Российской Федерации.

По заключению Росздравнадзора РФ № КП-20-006 от 14.04.2020г. Программный комплекс "Про родинки" является немедицинским программным обеспечением для неограниченного круга пользователей в образовательных, научно-популярных, справочно-информационных целях, в том числе для выбора медицинского специалиста.

Принцип пользования приложением «Про родинки»:

* Устанавливаете приложение;
* указываете особенности кожи и другие характеристики, влияющие на риск заболевания;
* фотографируете кожное образование камерой телефона по особым правилам:
  + на фотографии должен быть только один объект – «родинка»;
  + фотография должна быть резкой, так как чем меньше резкость – тем больше вероятность ошибки;
  + фотографируемый объект должен занимать наибольшую площадь кадра со всеми контурами объекта.
* система искусственного интеллекта на основе нейросетей глубокого обучения анализирует данные;
* по результатам анализа Приложение выдает рекомендацию о выборе врача;

### Платформа «Celsus»

ООО «Медицинские скрининг системы» основана в 2018 году и является разработчиком высокотехнологичных решений в области радиологии.

Решение «Цельс» - сложноорганизованная нейронная сеть, искусственный интеллект, система поддержки принятия врачебных решений на основе нейронных сетей и математического анализа путем анализа цифровых медицинских изображений, детектирования объектов и интерпретации результата.

«Цельс» является первым в РФ медицинским изделием с применением ИИ, зарегистрированным по 3 классу риска, также зарегистрированным в Евросоюзе.

«Celsus» может применяться по различным сценариям: начиная с использования на рабочем места врача при приеме пациента заканчивая автоматической потоковой обработкой поступающих данных.

Сейчас один из важных сценариев применения — массовый скрининг, который проводится в регионах, например, на передвижных маммографах или при проведении диспансеризации.

Анализируя изображения, «Celsus» находит и классифицирует патологические объекты (новообразования), кальцинаты, плотность ткани и иные параметры. А дальше присваивает класс по шкале Bi-RADS (европейский стандарт чтения и интерпретации снимков молочной железы), ранжирует итоговый список пациентов от наибольшего к меньшему по риску наличия патологии. Врач получает информацию о том, снимки какого пациента необходимо исследовать в первую очередь или отправить в онкоцентр в приоритетном порядке.

На данном этапе, Celsus готов к промышленной эксплуатации. Вышел первый релиз блока по выявлению патологий при проведении флюорографии. Команда работает над релизом по выявлению рака легких при рентгенографических исследованиях. В планах — анализировать данные компьютерной томографии (современный метод рентгеновского послойного исследования, более точный, чем рентгенография) и морфологии (исследования тканей организма, полученных путем пункции, биопсии или хирургически, на предмет наличия раковых клеток; самый точный метод подтверждения диагноза).

## Постановка цели и задачи

Чтобы произвести исследование результатов применения алгоритма приведения к средней яркости и выявить возможные закономерности, необходимо разработать программу, которая должна решать следующие задачи:

* Загрузка экземпляров изображений в базу данных;
* открытие изображений;
* преобразование яркости каждого пикселя изображения к средней яркости 127.5;
* выводить гистограммы яркостей изображения R, G, B;
* выводить значения средней яркости по гистограммам R, G, B;
* выводить значения медианы по гистограммам R, G, B;
* выводить значения среднеквадратичного отклонения по гистограммам R, G, B;
* сохранение данных исследования;
* загрузка данных исследования.

## Вывод по аналитической части

На данном этапе был проведен анализ предметной области, существующих на данный момент проблем и их последствий, а также были выполнены постановка цели и задач, решение которых необходимо для достижения поставленной цели.

Проанализировав положительные и отрицательные стороны языков программирования, можно сделать вывод, что C# является оптимальным языком программирования для реализации программы. Что касается среды разработки, выбор был между Eclipse и Visual Studio. Сравнение сред разработки и представлены в Таблица 1.

Таблица 1 - Сравнение сред Visual Studio и Eclipse

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Условия предоставления | Кроссплатформенность | Функциональность |
| Visual Studio | Бесплатна для малых предприятий, учащихся | Нацелено на .Net (Windows) | Необходимый функционал встроен в коробку, достаточно скачать необходимые плагины |
| Eclipse | Бесплатна | Поддерживает Windows, Linux, macOS | Необходимо скачать и настроить плагины для корректной работы |

Для разработки программы был выбран язык программирования C# из-за простоты написания программы и наличия библиотеки Marshal, которая позволила многократно ускорить обработку изображений. В качестве среды разработки была выбрана Visual Studio. С точки зрения кроссплатформенности нужно было выбрать Eclipse, но исходя из требований задания на ВКР, платформа Windows подходит наиболее лучше.

Используя Xamarin, можно выпустить программу и для Android, iOS, Windows Phone устройств.

# Проектная часть

## Описание предметной области

В соответствие с предметной областью, система строится с учетом следующих особенностей обращения с пользователем:

1. Можно подключаться к базе данных.
2. Можно загружать новые экземпляры изображений.
3. Возможно фильтровать и выбирать варианты из списка доступных изображений.
4. Проводить исследование доступными средствами.
5. Сохранять и загружать результаты исследований.

## Концептуальное проектирование

Концептуальное проектирование – сбор, анализ и редактирование требований к данным.

Неотъемлемой частью концептуального проектирования является построение UML-диаграмм. Далее будут приведены некоторые UML- диаграммы, которые используются в данной работе.

### Варианты использования

Диаграммы вариантов использования описывают функциональное назначение системы или то, что система должна делать. Разработка диаграммы преследует следующие цели:

1. определить общие границы и контекст моделируемой предметной области;
2. сформулировать общие требования к функциональному поведению проектируемой системы;
3. разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
4. подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть диаграммы вариантов использования состоит в следующем. Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Диаграмма вариантов использования может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов.

На диаграмме вариантов использования (Рисунок 1) показано как пользователь может взаимодействовать с программой и какие действия доступны. Основные функции – это открытие экземпляров изображений, обработка и сохранение результатов исследований, а также обратная связь в виде выдачи ошибок в ходе выполнения алгоритма.

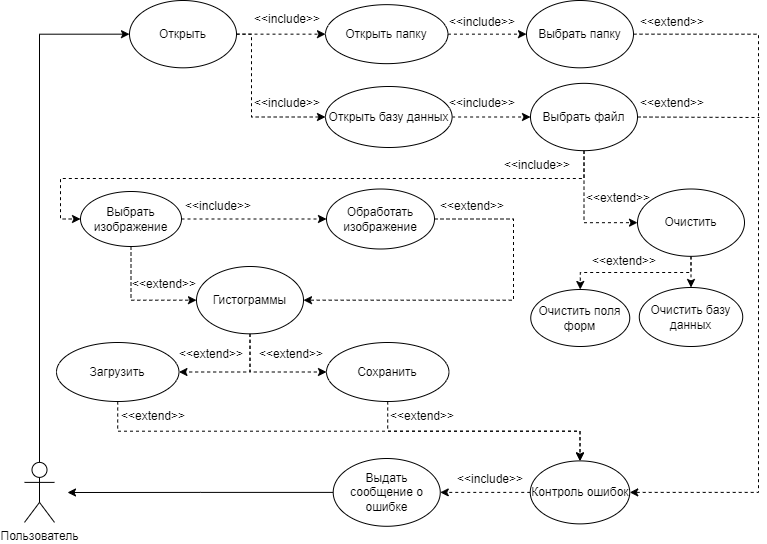


Рисунок 1 - Диаграмма вариантов использования пользователя

Спецификации диаграммы вариантов использования:

В роли актера выступает пользователь, который взаимодействует с программой.

Пользователю доступно действие открыть.

Для начала процесса обработки изображений, необходимо выполнить следующие шаги:

Пользователь – открыть базу данных – выбрать файл.

Чтобы выполнить загрузку экземпляров изображений в базу данных, необходимо выполнить следующие шаги:

Пользователь – открыть папку – выбрать папку.

Шаги, необходимые для исследования изображения:

Пользователь – открыть – открыть базу данных – выбрать файл – выбрать изображение – обработать изображение.

Шаги, необходимые для сохранения исследования:

Пользователь – открыть – открыть базу данных – выбрать файл – выбрать изображение – обработать изображение – гистограммы – сохранить.

Шаги, необходимые для загрузки исследования:

Пользователь – открыть – открыть базу данных – выбрать файл – выбрать изображение – обработать изображение – гистограммы – загрузить.

Шаги, необходимые для очистки данных из базы данных:

Пользователь – открыть – открыть базу данных – выбрать файл – очистить – очистить базу данных.

Шаги, необходимые для очистки рабочего пространства:

Пользователь – открыть – открыть базу данных – выбрать файл – очистить – очистить поля.

В программе также предусмотрена обратная связь, реализуемая выводом ошибок в ходе выполнения программы, чтобы вызвать сообщение достаточно смоделировать одну из следующих ситуаций:

* при загрузке исследования;
* при сохранении исследования;
* при открытии некорректного типа базы данных;
* при записи некорректного типа файла в базу данных.

### Схема базы данных

### Диаграмма классов

Центральное место в объектно-ориентированном программировании занимает разработка логической модели системы в виде диаграммы классов. Диаграмма классов (classdiagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений.

Диаграмма классов представляет собой граф, вершинами которого являются элементы типа «классификатор», связанные различными типами структурных отношений. Диаграмма классов может также содержать интерфейсы, пакеты, отношения и даже отдельные экземпляры, такие как объекты и связи.

На рисунке 2 приведена диаграмма классов разрабатываемой системы.

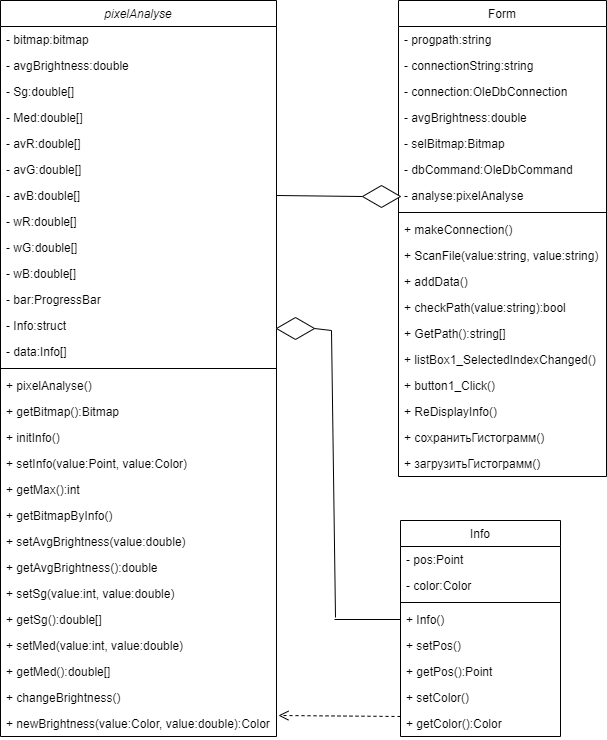


Рисунок 2 - Диаграмма классов

На рисунке 2 представлено изображение классов данной системы, а также соответствующих им операций и атрибутов.

Класс Form имеет атрибуты:

1. progpath;
2. connectionString;
3. connection;
4. avgBrightness;
5. selBitmap;
6. dbCommand;
7. analyse.

Класс Form имеет операции:

1. makeConnection();
2. ScanFile();
3. addData();
4. checkPath();
5. GetPath();
6. listBox1\_SelectedIndexChanged();
7. button1\_Click();
8. ReDisplayInfo();
9. сохранитьГистограмм();
10. загрузитьГистограмм().

Класс pixelAnalyse имеет атрибуты:

1. bitmap;
2. avgBrightness;
3. Sg;
4. Med;
5. avR;
6. avG;
7. avB;
8. wR;
9. wG;
10. wB;
11. bar;
12. data

Класс pixelAnalyse имеет операции:

1. pixelAnalyse();
2. getBitmap();
3. initInfo();
4. setInfo();
5. getMax();
6. getBitmapByInfo();
7. setAvgBrightness();
8. getAvgBrightness()
9. setSg();
10. getSg();
11. setMed();
12. getMed();
13. changeBrightness();
14. newBrightness().

## Блок-схема работы программы

Блок-схема — это диаграмма, на которой обычно представлен процесс, система или компьютерный алгоритм и которая используется для документирования, планирования, уточнения или визуализации многоэтапного рабочего процесса. Благодаря блок-схемам вы можете определить цели и масштабы рабочего процесса, а также установить необходимые задачи в хронологическом порядке. Впервые эта концепция была задокументирована в 1921 году, когда инженеры-технологи Фрэнк и Лилиан Гилбреты представили Американскому обществу инженеров-механиков (ASME) графическую схему технологического процесса. Популярность блок-схем росла на протяжении 20-х и 30-х годов, как и известность Арта Спинангера и Бена С. Грэхема, ставших приверженцами этой системы. Опираясь на результаты работы Гилбретов, в 1947 году Американское общество инженеров-механиков приняло систему символов для блок-схем. К 1949 году блок-схемы начали использовать при планировании компьютерных программ.

Ниже приведены некоторые наиболее часто встречающиеся символы блок-схем:

* Символ «Процесс», или «Действие»: этот символ обозначается прямоугольником и представляет собой этап процесса. Это самый распространенный символ блок-схемы.
* Символ «Начало/конец», или «Терминатор»: этот символ представляет собой прямоугольник с закругленными краями, который изображается как в начальной, так и в конечной точках блок-схемы.
* Символ «Решение»: ромбовидный символ, обозначающий вопрос, на который необходимо ответить «да»/«нет» или «правда»/«ложь». Затем блок-схема разбивается на различные ветви, представляющие собой отдельное направление для каждого ответа.
* Символ «Данные», или «Ввод/вывод данных»: символ в форме параллелограмма, означающий данные, которые вводятся в процесс или выводятся из него, то есть задействованные или полученные впоследствии ресурсы.

Рассмотрим работу алгоритма на блок-схемах (Рисунок 3-4), далее будут представлены две блок-схемы: загрузка изображения в базу данных, обработка изображения

## Выбор системы управлению базой данных

## Физическая модель базы данных

## Вывод по проектной части

# Технологическая часть

## Взаимодействие базы данных и программы

Взаимодействие базы данных и программы организовано при помощи технологии OleDB в C#.

OLE DB – набор COM-интерфейсов, которые позволяют приложениям унифицировано работать с данными разных источников и хранилищ информации. Разработана Microsoft в качестве дальнейшего развития технологии доступа к данным и должен был прийти на замену и в качестве преемника ODBC, расширяя набор функций для поддержки более широкого круга нереляционных источников данных, таких как объектно-ориентированные базы данных или электронные таблицы, и для которых не обязательно использовать SQL.

OLE DB отделяет хранилище данных от приложения, которое должно иметь доступ к нему через набор абстракций, состоящий из источника данных (DataSource), сессии (Session), команды (Command) и набора строк (Rowset). Это было сделано для предоставления унифицированного доступа к различным видам и источникам данных и изоляцию специфики взаимодействия с конкретным хранилищем. OLE DB концептуально разделена на потребителей (клиентов) и поставщиков (провайдеров). Потребителем является приложение, которому необходим доступ к данным, а поставщик реализует интерфейс доступа к данным и, следовательно, обеспечивает информацией потребителя.

Спецификация OLE DB является частью Microsoft Data Access Components (MDAC), представляющей собой группу технологий Microsoft, формирующих основу для единого и всеобщего способа разработки приложений для доступа к данным практически любого хранилища. В состав MDAC входят, к примеру, сервисы OLE DB (пул подключений и прочее) и компоненты ADODB.

Поскольку различные хранилища данных могут иметь разные возможности, поставщики OLE DB, как правило, не поддерживают все интерфейсы, описанные в спецификации OLE DB. Доступные возможности поставщика данных определяются через запрос указателей на COM интерфейсы его объектов или через чтение информационных свойств источника данных (DataSource). Поставщик данных также может реализовывать и предоставлять свои собственные COM-интерфейсы и структуры данных, не описанные в спецификации OLE DB.

Для записи и выборки данных о изображениях и исследованиях была выбрана база данных Microsoft Access.

К основным плюсам MS Access можно отнести:

* MS Access распространен, так как является продуктом Microsoft, программное обеспечение и операционные системы которой использует большая часть пользователей персональных компьютеров. MS Access полностью совместим с операционной системой Windows, постоянно обновляется производителем, поддерживает множество языков;
* MS Access ориентирован на людей с всесторонней профессиональной подготовкой по средствам наличия многих и простых инструментов работы с базой данных;
* возможность создавать базы данных и работать с ними, не зная языка запросов SQL, через диалоговые средства.

К основным минусам MS Access можно отнести:

* Ограничены возможности по обеспечению многопользовательской среды;
* в ранних версиях (до Access 2003) отсутствуют такие средства как триггеры и хранимые процедуры, что заставляет разработчиков возлагать поддержание бизнес-логики БД на клиентскую программу или разрабатывать процедуры с помощью встроенного средства VBA;
* обладает несложными способами защиты с использованием пароля БД (возможно применения дополнительных мер по защите от несанкционированного доступа с использованием процедур VBA);
* в вопросах поддержки целостности данных отвечает только моделям БД небольшой и средней сложности;
* не распространяется бесплатно.

В отличие от других настольных СУБД, Access хранит все данные в одном файле, хотя и распределяет их по разным таблицам, как и положено реляционной СУБД. К этим данным относится не только информация в таблицах, но и другие объекты базы данных, которые будут описаны ниже.

Для выполнения почти всех основных операций Access предлагает большое количество Мастеров (Wizards), которые делают основную работу за пользователя при работе с данными и разработке приложений, помогают избежать рутинных действий и облегчают работу не обученному в программировании пользователю.

Создание многопользовательской БД Access и получение одновременного доступа нескольких пользователей к общей базе данных возможно в локальной одноранговой сети или в сети с файловым сервером. Сеть обеспечивает аппаратную и программную поддержку обмена данными между компьютерами. Access следит за разграничением доступа разных пользователей к БД и обеспечивает защиту данных. При одновременной работе. Так как Access не является клиент серверной СУБД, возможности его по обеспечению многопользовательской работы несколько ограничены. Обычно для доступа к данным по сети с нескольких рабочих станций, файл БД Access (с расширением \*.mdb) выкладывается на файловый сервер. При этом обработка данных ведется в основном на клиенте – там, где запущено приложение, в силу принципов организации файловых СУБД. Этот фактор ограничивает использование Access для обеспечения работы множества пользователей (более 15–20) и при большом количестве данных в таблицах, так как многократно возрастает нагрузка не сеть.

## Используемые классы объектов

Класс OleDb – содержит классы, используемые для подключения к поставщику OLE DB, включая OleDbCommand, OleDbConnection и OleDbDataAdapter. Эти классы поддерживают большинство поставщиков OLE DB, но не те, что требуют интерфейсов OLE DB версии 2.5.

Класс DbConnection – позволяет подключаться к хранилищу данных и отключаться от него. Кроме того, объекты подключения обеспечивают доступ к соответствующим объектам транзакций.

Класс DbCommand – представляет SQL-запрос или хранимую процедуру. Кроме того, объекты команд предоставляют доступ к объекту чтения данных конкретного поставщика данных.

Класс DbDataReader – представляет SQL-запрос или хранимую процедуру. Кроме того, объекты команд предоставляют доступ к объекту чтения данных конкретного поставщика данных.

Класс Windows Forms – представляет окно или диалоговое окно, которое составляет пользовательский интерфейс приложения.

Form – это представление любого окна, отображаемого в приложении. Класс Form можно использовать для создания стандартных, инструментов, безграничных и плавающих окон. Класс также можно использовать для создания модальных окон, таких как диалоговое окно. Форма с несколькими документами (MDI) может содержать другие формы, называемые дочерними формами MDI. Форма MDI создается путем присвоения свойству IsMdiContainer значения true. Дочерние формы MDI создаются путем установки MdiParent свойства родительской формы MDI, которая будет содержать дочернюю форму.

Используя свойства, доступные Form в классе, можно определить внешний вид, размер, цвет и функции управления окнами создаваемого окна или диалогового окна. Свойство Text позволяет указать заголовок окна в строке заголовка. Свойства Size позволяют DesktopLocation определить размер и положение окна при его отображении. Свойство цвета можно использовать для ForeColor изменения цвета переднего плана по умолчанию для всех элементов управления, размещенных в форме. MinimizeBox, FormBorderStyle, MaximizeBox свойства позволяют управлять тем, может ли форма быть свернута, развернута или изменена во время выполнения.

Помимо свойств, можно использовать методы класса для управления формой. Например, можно использовать ShowDialog метод для отображения формы в виде модального диалогового окна. Метод можно использовать для SetDesktopLocation размещения формы на рабочем столе.

События Form класса позволяют реагировать на действия, выполняемые в форме. Событие можно использовать для Activated выполнения таких операций, как обновление данных, отображаемых в элементах управления формы при активации формы.

Форму можно использовать в качестве начального класса в приложении, поместив метод, вызываемый Main в классе. В методе Main добавьте код для создания и отображения формы. Кроме того, для запуска формы необходимо добавить STAThread атрибут Main в метод. При закрытии начальной формы приложение также закрывается.

Класс OpenFileDialog – отображает диалоговое окно, позволяющее пользователю открыть файл. Этот класс позволяет проверить, существует ли файл и открыть его. Свойство ShowReadOnly определяет, отображается ли в диалоговом окне флажок только для чтения. Свойство ReadOnlyChecked указывает, установлен ли флажок только для чтения.

Класс SaveFileDialog – запрашивает у пользователя местоположение для сохранения файла. Этот класс может открывать и перезаписывать существующий файл или создавать новый файл.

Класс Drawing – абстрактный класс, описывающий двумерный рисунок. Этот класс не может наследоваться кодом.

Drawing объекты — это легковесные объекты, позволяющие добавлять геометрические фигуры, изображения, текст и мультимедиа в приложение. Drawing объекты считаются легкими, так как они не поддерживают макет, обзор ввода и фокус. Благодаря преимуществам производительности рисунки идеально подходят для фона и картинки. При программировании на Visual уровне также используются чертежи.

Поскольку они наследуют от Freezable класса, Drawing объекты предоставляют дополнительные функции, которые делают их полезными для описания картинок и фона: их можно объявить, как ресурсы, совместно используемые среди нескольких объектов, сделать доступными только для чтения и сделать потокобезопасными. Дополнительные сведения о различных функциях, предоставляемых Freezable объектами.

Класс IO – cодержит типы, позволяющие осуществлять чтение и запись в файлы и потоки данных, а также типы для базовой поддержки файлов и папок.

Класс Path – выполняет операции для экземпляров класса String, содержащих сведения о пути к файлу или каталогу. Эти операции выполняются межплатформенным способом.

Класс Collections.Generic – содержит интерфейсы и классы, определяющие универсальные коллекции, которые позволяют пользователям создавать строго типизированные коллекции, обеспечивающие повышенную производительность и безопасность типов по сравнению с не универсальными строго типизированными коллекциями.

Класс pixelAnalyse – собственный класс, необходимый для хранения и обработки информации для текущего изображения в исследовании. К основным функциям этого класса относится следующее:

* Хранение информации о пикселях изображения (позиция на полотне и цвет RGB);
* хранение и расчет количества пикселей каждой яркости (0-255) всех трех шкал: R, G, B;
* хранение и расчет частот яркости (0-255) всех трех шкал: R, G, B;
* хранение и расчет значения средней яркости всего изображения;
* хранение и расчет значений среднеквадратичного отклонения и медианы;
* хранение и обработка экземпляра изображения Bitmap;
* обработка новой яркости.

## Описание работы программы

При запуске программы (Рисунок 1) необходимо соединиться с базой данных для доступа к экземплярам изображений (Рисунок 2).

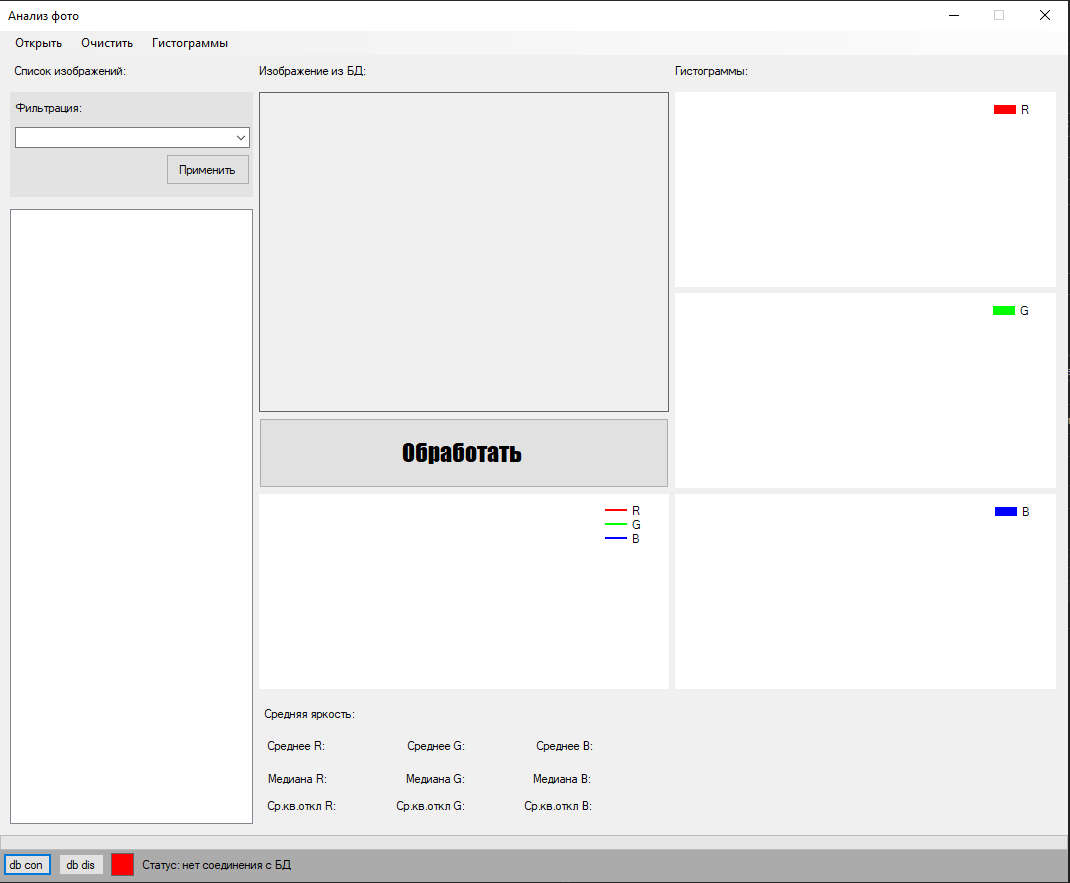


Рисунок 3 – Программа

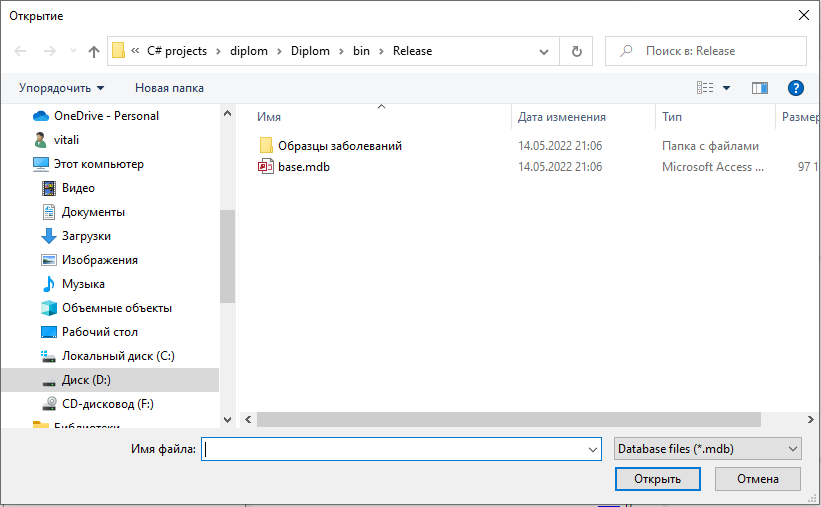


Рисунок 4 - Подключение к базе данных

Подключившись к базе данных, выводится список всех доступных изображений для исследования. Для более удобной навигации по списку, была реализована фильтрация по группе заболеваний (Рисунок 3).

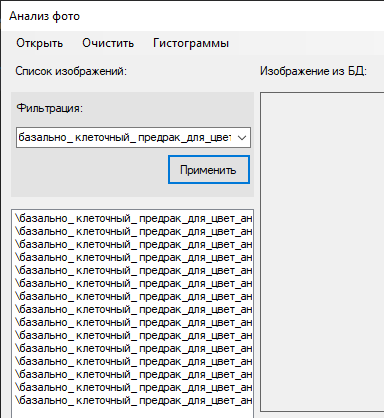


Рисунок 5 - Список изображений с фильтрацией

Нажав на имя изображения из списка – в область исследования картинок загружаются пиксели из базы данных (Рисунок 4).

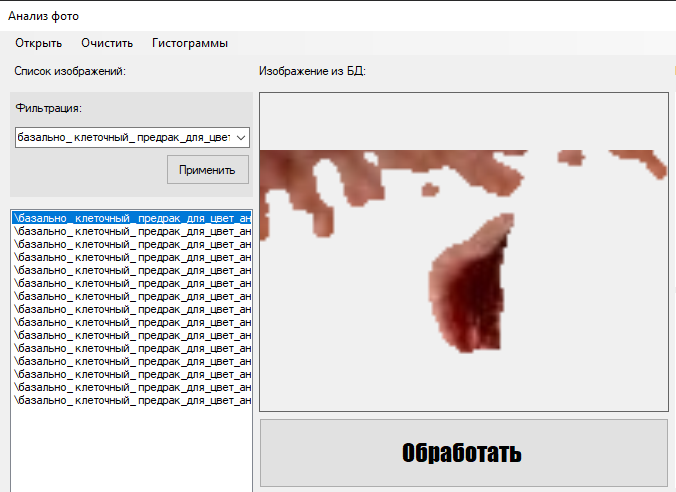


Рисунок 6 - Загруженное и базы данных изображение

После нажатия на кнопку обработать – происходит приведение всех пикселей к средней яркости 127,5 (Рисунок 5).

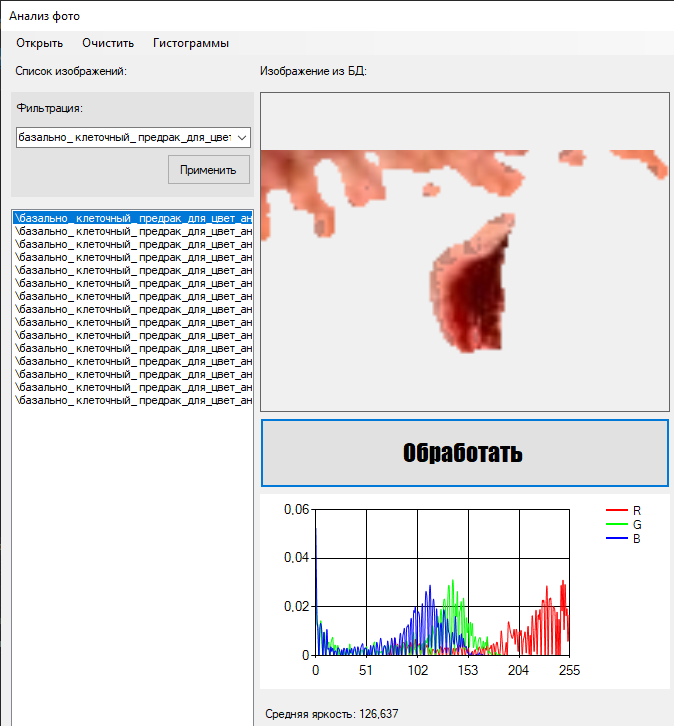


Рисунок 7 - Обработанное изображение

Загружая и проводя исследование над изображением – у него меняется яркость, а соответственно и частоты гистограмм R, G, B. Чтобы наглядно изучить данные – внизу есть информационно-графический блок (Рисунок 6), содержащий в себе следующие элементы:

* Значение средней яркости всего изображения;
* значения средней яркости R, G, B;
* значение параметра среднеквадратичного отклонения;
* значение параметра медианы;
* графики частот гистограмм R, G, B и совместный RGB.

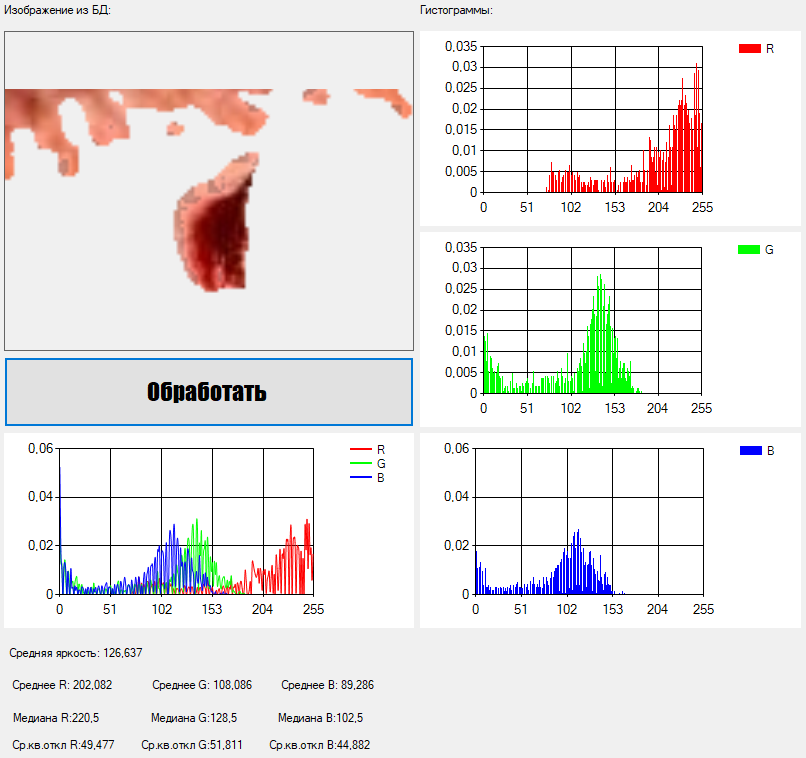


Рисунок 8 - Данные исследования

Добившись необходимых результатов по ходу изучения – существует возможность сохранить его для последующего открытия и просмотра (Рисунок 7).

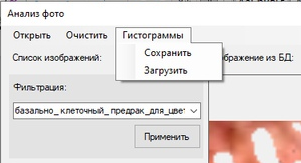


Рисунок 9 - Сохранение и загрузка исследования

Если возникла ситуация, когда необходимо использовать другие экземпляры изображений – это возможно сделать, если выбрать в контекстном меню пункт Очистить – Базу данных (Рисунок 8).

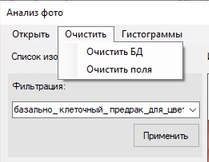


Рисунок 10 - Очистка базы данных и пользовательских полей ввода-вывода

Для обновления экземпляров изображений нужно выбрать в контекстном меню пункт Открыть – Папку (Рисунок 9). После некоторого времени, зависящего от количества и размера файлов, в базе данных будут записаны данные о названии изображения, диагнозе, информации о пикселях.

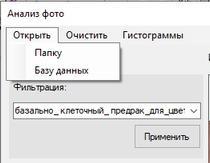


Рисунок 11 - Открытие новых экземпляров изображений

## Вывод по технологической части

На данном этапе были рассмотрены взаимодействие базы данных и программы, используемые классы объектов и их поведение в рамках разрабатываемой информационной системы.

# Исследовательская часть

## Входные данные

## Данные и инструменты анализа

## Вывод по технологической части

Список используемой литературы

*1. Приоров, А.Л.*Цифровая обработка изображений: учебное пособие / А.Л. Приоров, И.В. Апальков, В.В. Хрящев; Яросл. ос. ун-т. – Ярославль: ЯрГУ, 2007. – 235 с*.*

2. Матвеев В.А., Раухваргер А.Б. Исследование возможности использования цветной модели RGB для автоматизации спектрального анализа. / Сб. материалов конф. В 2 ч. Ч. 1 [Электронный ресурс]. – Ярославль: Издательство ЯГТУ, 2021. с 659-661.

3. МЕДСИ // Базальноклеточный рак (базалиома). [Электронный ресурс] <https://spb.medsi.ru/articles/bazalnokletochnyy-rak-bazalioma/> (дата обращения 10.03.2022).