Система биометрического считывателя.

Программное обеспечение для ПК.

Описание программы.

РОФ.МГТУ.00001 04

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. *N*дубл. |  |
| Взам. Инв. *N* |  |
| Подп. и дата |  |
| Инв. *N* подп. |  |

Листов 17

Проверил

Рафиков А. Г.

\_\_\_. \_\_\_. 20\_\_\_

Разработал

*­­­*

Пушкин Н. Д.

\_\_\_. \_\_\_. 20\_\_\_

2023

Оглавление

[1 Аннотация 3](#_Toc156900392)

[2 Основная часть 4](#_Toc156900393)

[1. 4](#_Toc156900394)

[2. 4](#_Toc156900395)

[2.1. Общие сведения 4](#_Toc156900396)

[2.1.1. Обозначение и наименование программы 4](#_Toc156900397)

[2.1.2. Программное обеспечение, необходимое для функционирования программы 4](#_Toc156900398)

[2.1.3. Языки программирования 4](#_Toc156900399)

[2.2. Функциональные назначения 5](#_Toc156900400)

[2.2.1. Классы решаемых задач 5](#_Toc156900401)

[2.2.2. Назначение программы 5](#_Toc156900402)

[2.2.3. Сведения о функциональных ограничениях на применение 5](#_Toc156900403)

[2.3. Описание логической структуры 5](#_Toc156900404)

[2.3.1. Алгоритм программы 5](#_Toc156900405)

[2.3.2. Описание логической структуры 5](#_Toc156900406)

[2.3.3. Алгоритм программы 6](#_Toc156900407)

[2.3.4. Используемые методы 6](#_Toc156900408)

[2.3.5. Структура программы с описанием функций составных частей и связи между ними 14](#_Toc156900409)

[2.3.6. Связи программы с другими программами 14](#_Toc156900410)

[2.3.7. Используемые технические средства 14](#_Toc156900411)

[2.4. Вызов и загрузка 14](#_Toc156900412)

[2.4.1. Способ вызова программы с соответствующего носителя данных 14](#_Toc156900413)

[2.4.2. Входные точки в программу 15](#_Toc156900414)

[2.5. Входные данные 15](#_Toc156900415)

[2.5.1. Характер, организация и предварительная подготовка входных данных 15](#_Toc156900416)

[2.5.2. Формат, описание и способ кодирования входных данных 15](#_Toc156900417)

[2.6. Выходные данные 15](#_Toc156900418)

[2.6.1. Характер и организация выходных данных 15](#_Toc156900419)

[2.6.2. Формат, описание и способ кодирования выходных данных 16](#_Toc156900420)

# 1 Аннотация

Конечная работа посвящена проектированию и разработке биометрического считывателя. Исследование включает в себя изучение основных принципов и технологий, лежащих в основе современных биометрических систем, с последующим созданием и описанием концепции считывателя. Работа представляет процесс проектирования, включая выбор биометрических параметров для идентификации, разработку аппаратных и программных компонентов, а также тестирование и оценку эффективности разработанного устройства.

# 2 Основная часть



## Общие сведения

## Обозначение и наименование программы

Наименования программы: биометрический считыватель.

## Программное обеспечение, необходимое для функционирования программы

Для работы программы необходимо определенное программное окружение, обеспечивающее минимальные требования к ресурсам.

Требования к составу периферийных устройств заключаются в наличии локальной вычислительной сети, что обеспечивает эффективное взаимодействие программы с другими устройствами и программным обеспечением, участвующими в процессе биометрической идентификации.

Программа не накладывает конкретных требований на параметры периферийных устройств, что позволяет ей гибко адаптироваться к различным типам оборудования без дополнительной настройки или модификации.

Относительно операционной системы, программа может успешно функционировать на ОС семейства GNU/Linux или Windows, что обеспечивает широкую совместимость с большинством существующего оборудования.

Для стабильной работы программы рекомендуется наличие не менее 1 Гб оперативной памяти на выделенном сервере, что обеспечит эффективную обработку данных и выполнение функций программы без существенных задержек или снижения производительности.

## Языки программирования

Программа написана на языке программирования Python.

## Функциональные назначения

## Классы решаемых задач

Программа разработана для решения задач биометрической идентификации. Ее основная задача состоит в надежном и точном определении личности путем сопоставления предоставленных биометрических данных с сохраненными образцами в базе данных.

## Назначение программы

Программа предназначена для использования в системах безопасности, доступа к конфиденциальной информации, а также в других областях, где требуется точная идентификация личности. Ее основная цель - обеспечить безопасный и удобный доступ к объектам, информации или системам, используя уникальные биометрические данные вместо традиционных методов аутентификации, таких как пароли или ключи.

## Сведения о функциональных ограничениях на применение

Программа имеет ряд ограничений на применение, включая требования к аппаратному и программному обеспечению. Ее функционирование зависит от наличия локальной вычислительной сети для взаимодействия с другими устройствами. Она также ограничена поддержкой операционных систем семейства GNU/Linux или Windows. Для эффективной работы рекомендуется выделенный сервер с не менее чем 1 Гб оперативной памяти.

## Описание логической структуры

## Алгоритм программы

Алгоритм программы приведен на отдельном документе «Схема алгоритма».

## Описание логической структуры

Логически программа делится на 3 части:

1. Фоновая камера. Функционал камеры ограничен ее возможности, без возможности использования ПО сервера.
2. Регистрация. Процесс записи видеоматериала с биометрическими данными.
3. Аутентификация. Процесс сопоставления видеопотока с камер с изображением человека с раннее созданной базой зарегистрированных пользователей.

## Алгоритм программы

Алгоритм программы приведен на отдельном документе «Текст программы».

## Используемые методы

Далее приведены разработанные методы:

* check\_liveness – проверка на liveness изображения с камеры.
  + Входные данные:
    - eyes\_count (int) – число найденных глаз (закрытых и открытых) фильтров Хаара.
    - open\_eyes\_count (int) – число найденных открытых глаз фильтров Хаара.
  + Выходные данные: True или False (в зависимости от близости входных параметров)
  + Переменные метода: отсутствуют
* add\_registration\_log – создание записи в журнал событий регистраций.
  + Входные данные:
    - log\_index (int) – номер события.
    - name (string) – имя регистрирующего пользователя.
  + Выходные данные: отсутствуют
  + Переменные метода:
    - Messages (list) – список возможных событий.
    - Now (date) – текущее время.
    - formatted\_date (date) – текущее время с указанием года и месяца.
    - File (file) – файл, куда записываются события.
* add\_auth\_log – создание записи в журнал событий аутентификации.
  + Входные данные:
    - log\_index (int) – номер события.
    - name (string) – имя аутентифицирующего пользователя.
  + Выходные данные: отсутствуют
  + Переменные метода:
    - Messages (list) – список возможных событий.
    - Now (date) – текущее время.
    - formatted\_date (date) – текущее время с указанием года и месяца.
    - File (file) – файл, куда записываются события.open\_back\_camera – включение фоновой камеры.
* open\_back\_camera – метод включения фоновой камеры.
  + Входные данные:
    - Id (int) – идентификатор камеры.
  + Выходные данные: отсутствуют.
  + Переменные метода:
    - Ret (bool) – результат получения изображения с камеры.
    - Camera (opencv2.camera) – получение камеры.
    - frame (opencv2.frame) – кадр изображения с камеры.
    - dim (tuple) - размер изображения.
* make\_video – запись видео при регистрации.
  + Входные данные:
    - Id\_camera (int) – идентификатор камеры.
    - name (string) – имя регистрирующего пользователя.
  + Выходные данные: index, liveness
    - liveness (bool) – результат проверки на liveness.
    - index (int) – номер (название) видео.
  + Переменные метода:
    - cascade\_path, eye\_cascade, righteye\_cascade (string) – пути до фильтров (лица, открытых глаз, открытых и закрытых глаз).
    - all\_eyes\_count, eyes\_count (int) – количество найденных глаз (открытых и закрытых, открытых).
    - Eyes, allEyes, Classifier (opencv2.CascadeClassifier) –классификаторы, полученные из фильтров Хаара.
    - Camera (opencv2.camera) – получение камеры.
    - start\_time, delta\_time\_seconds, current\_time (datetime.datetime) – переменные времени начала и длительности работы и текущего.
    - frame\_width, frame\_height, size (int) – размеры изображения.
    - Index (int) – номер (название) видео.
    - Fourcc (cv2.VideoWriter.fourcc) – формат видеофайла.
    - video\_file (cv2.VideoWriter) – видеофайл.
    - liveness (bool) – результат проверки на liveness.
    - Ret (bool) – результат получения изображения с камеры.
    - frame (opencv2.frame) – кадр изображения с камеры.
    - Gray (opencv2.cvtColor) – контрастность изображения.
    - Faces (classifier.detectMultiScale) – найденные лица.
    - x, y, w, h, ex, ey, ew, eh (int) – координаты изображения.
    - roi\_gray (classifier.detectMultiScale), roi\_color (opencv2.frame) – часть кадра и его контрастность.
* make\_photos\_from\_video – создание фото из видео при регистрации с последующем сохранением в базу данных.
  + Входные данные:
    - Name (string) – имя пользователя.
    - video\_number (int) – номер видео.
  + Выходные данные: отсутствуют.
  + Переменные метода:
    - count (int) – номер фото.
    - cap (opencv2.camera) – получение изображения.
    - frame (opencv2.frame) – кадр изображения с камеры.
    - Ret (bool) – результат получения изображения с камеры.
    - Fps (CAP\_PROP\_FPS), multiplier (int) - частота изображения и ее множитель.
    - frame\_id (opencv2.camera.frame) – фото из видео.
    - K (opencv2.waitKey) – переменная распознавания нажатия на клавиатуру.
* do\_authentification – метод проведения сравнений видеопотока и базы данных регистраций.
  + Входные данные:
    - id\_camera (int) – идентификатор камеры.
  + Выходные данные:
    - Result (bool) – результат распознавания.
    - name (string) – имя пользователя.
    - Liveness (bool) – результат проверки на Liveness (“человечность”).
  + Переменные метода:
    - cascade\_path, eye\_cascade, righteye\_cascade (string) – пути до фильтров (лица, открытых глаз, открытых и закрытых глаз).
    - all\_eyes\_count, eyes\_count (int) – количество найденных глаз (открытых и закрытых, открытых).
    - Eyes, allEyes, Classifier (opencv2.CascadeClassifier) –классификаторы, полученные из фильтров Хаара.
    - Camera (opencv2.camera) – получение камеры.
    - camera\_is\_ready (bool) – результат получения изображения с камеры.
    - frame (opencv2.frame) – кадр изображения с камеры.
    - Gray (opencv2.cvtColor) – контрастность изображения.
    - faces\_location (face\_recognition.face\_locations) – положение лиц.
    - faces\_encodings (face\_recognition.face\_encodings) – декодирование изображения лиц в вектор.
    - Faces (classifier.detectMultiScale) – найденные лица.
    - x, y, w, h, ex, ey, ew, eh (int) – координаты изображения.
    - roi\_gray (classifier.detectMultiScale), roi\_color (opencv2.frame) – часть кадра и его контрастность.
* do\_authentification\_with\_name – метод проведения сравнений видеопотока и базы данных регистраций с выведением имени найденного человека.
  + Входные данные:
    - id (int) – идентификатор камеры.
  + Выходные данные: отсутствуют.
  + Переменные метода:
    - Result (bool) – результат распознавания.
    - name (string) – имя пользователя.
    - Liveness (bool) – результат проверки на Liveness (“человечность”).
    - unknown\_name (string) – переменная имени неизвестного пользователя.
    - name (string) – переменная имени найденного пользователя.
    - cascade\_path, eye\_cascade, righteye\_cascade (string) – пути до фильтров (лица, открытых глаз, открытых и закрытых глаз).
    - start\_time, delta\_time\_seconds, current\_time (datetime.datetime) – переменные времени начала и длительности работы и текущего.
    - all\_eyes\_count, eyes\_count (int) – количество найденных глаз (открытых и закрытых, открытых).
    - Eyes, allEyes, Classifier (opencv2.CascadeClassifier) –классификаторы, полученные из фильтров Хаара.
    - Camera (opencv2.camera) – получение камеры.
    - \_ (bool) – результат получения изображения с камеры.
    - frame (opencv2.frame) – кадр изображения с камеры.
    - Gray (opencv2.cvtColor) – контрастность изображения.
    - Faces (classifier.detectMultiScale) – найденные лица.
    - x, y, w, h, ex, ey, ew, eh (int) – координаты изображения.
    - roi\_gray (classifier.detectMultiScale), roi\_color (opencv2.frame) – часть кадра и его контрастность.
* register\_another\_person – метод регистрации.
  + Входные данные:
    - id\_camera (int) – идентификатор камеры.
  + Выходные данные: отсутствуют.
  + Переменные метода:
    - name (string) – имя пользователя.
    - video\_path (str) – местоположение, куда запишем видео.
    - Content (list), file (str) – содержимое папки video\_path.
    - name\_is\_exist (bool) – переменная проверка наличия имени.
    - Index - номер (название) видеофайла.
    - Liveness (bool) – результат проверки на Liveness (“человечность”).
* add\_photo\_to\_db - Добавление в базу данных векторов лиц.
  + Входные данные:
    - Name (string) – имя пользователя.
    - Index (int) – номер видео.
    - Connection (sqlite3.connect) – подключение к базе данных.
    - Cursor (sqlite3.connect.cursor) – переменная управления базой данных.
    - Images, image (list, str) – временные изображения из видеопотока регистрации.
    - faces\_img (face\_recognition.face\_locations) – положение лиц.
    - faces\_enc (face\_recognition.face\_encodings) – декодирование изображения лиц в вектор.
    - face\_pickle (blob) – сериализированный вектор.
  + Выходные данные: отсутствуют.
  + Переменные метода:
    - secret.encryption\_key (string) – ключ шифрования.
* compare\_face\_with\_db – Сравнение лиц из видепотока и базы данных.
  + Входные данные:
    - face\_from\_camera (face\_recognition.face\_encodings) – декодирование изображения лиц в вектор.
  + Выходные данные:
    - Result – результат сравнения.
    - User[0] – имя пользователя.
  + Переменные метода:
    - users (list), user (list) – записи пользователей в базе данных.
    - face\_pickle (blob) – сериализированный вектор.
* main – Функция запуска программы.
  + Входные данные: отсутствуют
  + Выходные данные: отсутствуют.
  + Переменные метода:
    - mode (string) – режим работы программы.

## Структура программы с описанием функций составных частей и связи между ними

Алгоритм программы приведен на отдельном документе «Схема алгоритма».

## Связи программы с другими программами

Программа имеет следующие зависимости:

* Datetime – встроенная библиотека времени.
* Os – встроенная библиотека для работы с операционной системы. Используется для работы с файлами.
* time – встроенная библиотека времени.
* cv2 – модуль opencv2 для компьютерного зрения.
* face\_recognition – зависимость opencv2 для работы с лицами людей.
* sys – встроенная библиотека для чтения параметров запуска программы.
* numpy – библиотека математических операций.
* sqlite3 – библиотека для работы с БД SQLite.
* Pickle – библиотека для сериализации.
* pysqlcipher3 – библиотека для шифрования базы данных.

## Используемые технические средства

Программа использует IP-камеры, видеопроцессор hi3520 и ОС Windows/ Unix на стороне серверного компонента.

## Вызов и загрузка

## Способ вызова программы с соответствующего носителя данных

Способ вызова приведен в отдельных документах «Руководство пользователя», «Руководство администратора».

Или собрать проект самостоятельно. Для сборки проекта написан файл CmakeLists.txt

С помощью команды cmake <папка с исходными текстами> необходимо сгенерировать файлы для компиляции.

С помощью команды make собрать проект.

Также проект можно собрать с помощью утилиты pyinstaller.

1. Скачаем утилиту для сборки с помощью команды pip install PyInstaller.
2. Необходимо скачать зависимости проекта, чтобы утилита добавила их в проект. Зависимости можно скачать с использованием команды pip install -r requirements.txt
3. Утилита имеет два варианта сборки: команда pyinstaller --onefile biometrics.py создаст 1 исполняемый файл (\*.exe), вложив в него зависимости. pyinstaller biometrics.py создаст исполняемый файл (\*.exe) с отдельной библиотекой зависимостей в виде dll-файлов.

## Входные точки в программу

Входные точки приведены в отдельном документе «Руководство администратора».

## Входные данные

## Характер, организация и предварительная подготовка входных данных

Входными данными являются номер используемых камер, режим работы и передаваемый далее видеопоток с камер.

Входные данные не требуют предварительной подготовки.

## Формат, описание и способ кодирования входных данных

Формат, описание и способ кодирования приведены в отдельном документе «Руководство администратора».

## Выходные данные

## Характер и организация выходных данных

Выходные данные представляют собой сообщение в консоль о результате выполнения.

## Формат, описание и способ кодирования выходных данных

Формат, описание и способ кодирования приведены в отдельном документе «Руководство администратора».

Выходные данные не требуют дальнейшей организации.

**Лист регистрации изменений к РКД**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Содержание изменения / основание / дата внесения изменения | Количество  страниц РКД | Подпись автора РКД |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |