

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

|  |  |  |
| --- | --- | --- |
|  | | |
| зав. кафедрой | | «КБИС» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | О.А. Сафарьян |
| (подпись) | |  |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. | |

**ОТЧЕТ**

по производственной, проектно-технологической практике

в ФГАНУ НИИ «Спецвузавтоматика»

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. П. Ковалев

подпись, дата

Обозначение отчета УП.990000.000 Группа ВКБ32

|  |  |
| --- | --- |
| Специальность 10.05.01 | Компьютерная безопасность |
| Специализация | Математические методы защиты информации |

Руководитель практики:

|  |  |  |  |
| --- | --- | --- | --- |
| от предприятия | зам. директора по научной работе | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | К. Ю. Гуфан |
| от кафедры | доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | О. С. Бурякова |

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись преподавателя

Ростов-на-Дону

2025 г.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

**ЗАДАНИЕ**

по производственной, проектно-технологической практике

в ФГАНУ НИИ «Спецвузавтоматика»

в период с «10» июня 2025 г. по «6» июля 2025 г.

Обучающийся Ковалев Данил Петрович

Обозначение отчета УП.990000.000 Группа ВКБ32

Срок представления отчета на кафедру «\_\_» \_\_\_\_\_\_\_\_ 2025 г.

Содержание индивидуального задания

1. Телеграмм-бот “Помогатор” – инструмент для творчества;
2. оформление списка использованных информационных ресурсов по ГОСТ Р 7.0.100–2018;
3. оформление отчета по практике согласно документу “Правила оформления письменных работ обучающихся для технических направлений подготовки” от 16.12.2020 приказа 242 “О введение документов действие”.

|  |  |  |
| --- | --- | --- |
| Руководитель практики от  кафедры | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | О. С. Бурякова |
| Руководитель практики от  предприятия | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | К. Ю. Гуфан |
| Задание принял к исполнению | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Д. П. Ковалев |



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника

Кафедра «Кибербезопасность информационных систем»

|  |  |  |
| --- | --- | --- |
|  | | |
| И. о. зав. кафедрой | | «КБИС» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | О.А. Сафарьян |
| (подпись) | |  |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. | |

**Рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| **№** | **Мероприятие** | **Срок выполнения** |
| 1 | Прохождение вводного и первичного инструктажа по охране труда на рабочем месте, и инструктажа по пожарной безопасности на объекте. | 10.06.2025 |
| 2 | Получение индивидуального задания. | 10.06.2025 |
| 3 | Ознакомление с теоретическим материалом. | 11.06.2025-15.06.2025 |
| 4 | Оформление программной реализации. | 16.06.2025-29.06.2025 |
| 5 | Оформление отчёта по практике. | 29.06.2025-02.07.2025 |
| 6 | Защита отчёта по практике на предприятии. | 03.07.2025 |
| 7 | Защита отчёта по практике на кафедре. | 04.07.2025-06.07.2025 |

Руководитель практики:

|  |  |  |  |
| --- | --- | --- | --- |
| от предприятия | зам. директора по научной работе | \_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | К. Ю. Гуфан |
| от кафедры | доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | О. С. Бурякова |

Ростов-на-Дону

2025 г.

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

В данном разделе ежедневно, кратко и четко записываются выполняемые работы, и в конце каждой недели журнал представляется для проверки руководителю (от предприятия и университета) практики. При выполнении одной и той же работы несколько дней, в графе «дата» сделать запись «с \_\_\_по\_\_\_».

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Место работы | Выполняемые работы | Оценка руководителя |
| 10.06.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Знакомство с предприятием, прохождение вводного инструктажа. |  |
| 10.06.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Ознакомление с территорией предприятия, прохождение первичного инструктажа по ТБ, ПБ |  |
| 10.06.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Получение индивидуального задания. |  |
| 11.06.2025-15.06.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Ознакомление с теоретическим материалом |  |
| 17.06.2025-29.06.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Изучение общих характеристик методов сжатия информации. |  |
| 01.07.2025-02.07.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Оформление отчёта по практике. |  |
| 03.07.2025 | ФГАНУ НИИ «Спецвузавтоматика» | Защита отчёта по практике на предприятии. |  |
| 04.07.2025-06.07.2025 | кафедра «КБИС» | Защита отчёта по практике на кафедре. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| от предприятия | зам. директора по научной работе | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | К. Ю. Гуфан |
| от кафедры | доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | О. С. Бурякова |

ОТЗЫВ-ХАРАКТЕРИСТИКА

Обучающийся Ковалев Данил Петрович

группа ВКБ32

кафедра «Кибербезопасность информационных систем»

вид практики производственная практика

наименование места практики ФГАНУ НИИ «Спецвузавтоматика»

Обучающийся выполнил задания программы практики:

изучил задание, собрал информацию, разработал структуру, организацию и управление проекта, реализовал код, протестировал его, изготовил готовый продукт.

Дополнительно ознакомился/изучил:

Сверточные нейронные сети, разработка телеграмм ботов под Python с использованием aiogram.

Заслуживает оценки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Руководитель практики  от предприятия  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_ г.  М.П. |
|  |  |

**Содержание**

[Введение 6](#_Toc169561395)

[1 Криптографический алгоритм RSA 8](#_Toc169561396)

[1.1 История 9](#_Toc169561397)

[1.2 Вводные данные 11](#_Toc169561398)

[1.3 Описание 13](#_Toc169561399)

[1.4 Критическая оценка криптостойкости криптоалгоритма RSA 14](#_Toc169561400)

[1.5 Применение 18](#_Toc169561401)

[1.6 Модификации RSA 18](#_Toc169561402)

[2 Программная реализация 20](#_Toc169561403)

[2.1 Высокоуровневый язык программирования Python 20](#_Toc169561404)

[2.2. Общая информация про программную реализацию 20](#_Toc169561405)

[2.3. Конфигурационные файлы 21](#_Toc169561410)

[2.4. Логика создания ключей 24](#_Toc169561411)

[Заключение 28](#_Toc169561412)

[Перечень использованных информационных ресурсов 29](#_Toc169561413)

[Приложение А Техническое задание 30](#_Toc169561414)

[Приложение Б Листинг 1 34](#_Toc169561415)

# Введение

Своим существованием Федеральное государственное автономное научное учреждение «Спецвузавтоматика» обязано инициативному изобретателю Алексею Аграновскому, выпускнику физического факультета Ростовского государственного университета, который с 1975 года работал в университетском вычислительном центре, начав с техника и закончив в должности начальника отдела автоматизации научных исследований. В 1992 году он основал на университетской базе конструкторское бюро «Спецвузавтоматика», которое в 2002 году превратилось во ФГАНУ НИИ «Специализированные вычислительные устройства защиты и автоматика» при Минобрнауки РФ. Аграновский оставался директором института до своей смерти в 2010 году.

Целью деятельности ФГАНУ НИИ «Спецвузавтоматика» является   
создание и внедрение информационных технологий для развития инновационного потенциала различных государственных структур по решению их собственных задач в области науки, техники и образования.

Сфера компетенции ФГАНУ НИИ «Спецвузавтоматика» достаточно обширна, так как она включает в себя все аспекты работы в сферах   
безопасности и защиты информации, управления проектами и внедрения инновационных технологий. Также институтом проводятся различные научные исследования в области телекоммуникаций, создания технических средств  
связи, защиты и обработки информации, разрабатывается программное обеспечение по заказу и создаются электронные устройства, начиная от идеи и заканчивая серийными образцами.

В Институте создана организационная структура, позволяющая развивать прикладные исследования и научные школы, осваивать новые технологии, реализовывать инновационные продукты.

Существует несколько подразделений:

1. Научно-исследовательские лаборатории (НИЛ);
2. Южный региональный аттестационный центр (ЮРАЦ);
3. Центр трансфера технологий и коммерциализации (ЦТТК);
4. Центр космических технологий «Арктурус» (ЦКТ);
5. Национальный центр противодействия терроризму и экстремизму в образовательной среде и сети Интернет (НЦПТИ);

И кроме всего прочего существуют такие приоритетные направления деятельности:

1. выполнение работ и исследований в сфере информационных технологий, естественных и технических наук, информационной безопасности, инновационным методам разработки программного обеспечения, проблематике информационного противодействия терроризму и экстремизму в сети Интернет и образовательной среде, телекоммуникациям и управлению, созданию новых образовательных технологий;
2. координация научных исследований по перспективным направлениям области безопасности Российской Федерации;
3. разработка и внедрение в сферу образования перспективных системных

и обучающих технологий;

1. организация инновационной деятельности с учетом основных направлений социально-экономического развития Российской Федерации, реализации приоритетных направлений государственной политики Российской Федерации в научно-технической сфере;
2. внедрение в производство новых технологий и техники по заказам заинтересованных предприятий и организаций;
3. содействие развитию кадрового потенциала региона, в том числе в инновационной и проектной деятельности.

## 1 Актуальность темы

В современной цифровой среде компании и специалисты сталкиваются с необходимостью оперативно обрабатывать большие объемы данных, генерировать творческие решения и выделяться среди конкурентов. Дизайнеры, например, тратят значительное время на рутинные задачи, такие как подбор цветовых схем, создание макетов или анализ трендов. Одновременно бизнесу требуется эффективно привлекать целевую аудиторию, создавая персонализированный контент и стратегии продвижения, что требует глубокого понимания поведения потребителей.

Развитие искусственного интеллекта открывает возможность создания универсальных инструментов, способных интегрировать передовые нейросетевые технологии для решения указанных проблем. Системы, объединяющие интерфейсы с крупными языковыми моделями (LLM) вроде OpenAI, DeepSeek, а также внутренними специализированными моделями, позволяют автоматизировать рутинные процессы, генерировать идеи и анализировать данные в реальном времени.

### Обзор предметной области

Рассмотрим различные области, где может пригодиться текущий проект. В самом начале - применение в дизайне.

Современные дизайнеры сталкиваются с множеством рутинных задач, требующих времени и творческого подхода. Например:

* Генерация идей: Подбор цветовых палитр, стилей, шрифтов, соответствующих трендам;
* Создание прототипов: Быстрая разработка макетов интерфейсов, логотипов или рекламных материалов;
* Анализ конкурентов: Визуальная оценка продуктов конкурентов, выявление популярных дизайнерских решений.
* Генерация изображений по текстовым запросам (например, «создать логотип в стиле минимализма»);
* Персонализация решений под предпочтения клиента на основе исторических данных.

Такой подход снижает барьер входа для начинающих дизайнеров, ускоряет выполнение задач и позволяет фокусироваться на креативной части работы.

Теперь рассмотрим применение в маркетинге и привлечении аудитории. Эффективное продвижение требует глубокого понимания поведения целевой аудитории, персонализации контента и прогнозирования трендов. Современные маркетологи сталкиваются с задачами:

* Создание контента: Генерация текстов для рекламы, постов в соцсетях, email-рассылок;
* Сегментация аудитории: Анализ данных для формирования персонализированных предложений;
* SEO и SMM: Оптимизация ключевых слов, планирование публикаций, анализ эффективности кампаний.

### Постановка задачи

Целью проекта является создание продукта, позволяющего автоматизировать рутинные вещи для дизайнеров. Данный проект создан на основе микросервисной архитектуры, программный продукт обеспечивает унифицированный доступ к различным нейросетевым моделям (OpenAI, DeepSeek, собственная модель) и предоставляет собственные инструменты для автоматизации задач в сферах дизайна и маркетинга. Проект включает реализацию Telegram-бота как интерфейса взаимодействия с пользователем, что повышает доступность и удобство использования.

Разрабатываемое приложение должно содержать следующий функционал:

* Общение LLM (DeepSeek, OpenAI);
* Стилизация изображения;
* Изменение размеров изображения;
* Сжатие изображения;
* Поворот изображения на заданное количество градусов;
* Инверсия цвета изображения;
* Телеграм-бот, выступающий как посредник работы с нашим сервисом.

Для достижения поставленной цели необходимо решить задачи:

* + Разработать необходимые алгоритмы;
  + Выбрать и интегрировать модели машинного обучения;
  + Выбрать язык программирования;
  + Выбрать архитектуру и дизайн приложения;
  + Выбрать подходящие технологии и библиотеки;
  + Разработать удобный и интуитивно понятный для пользователя интерфейс;
  + Разработать приложение;
  + Протестировать приложение.

### 1.3 Модель VGG19

Модель переноса стиля изображения представляет собой революционный алгоритм в области компьютерного зрения, который позволяет объединить содержание одного изображения со стилистическими особенностями другого. Данная технология, впервые представленная в работе "Image Style Transfer Using Convolutional Neural Networks", открыла новый подход к созданию художественных изображений с помощью искусственного интеллекта.

В основе модели лежит использование предварительно обученной сверточной нейронной сети, изначально разработанной для задач классификации изображений. Сеть состоит из последовательности сверточных слоев (обозначенных как conv1\_1, conv1\_2, conv2\_1 и так далее), которые постепенно извлекают все более сложные и абстрактные характеристики изображения.

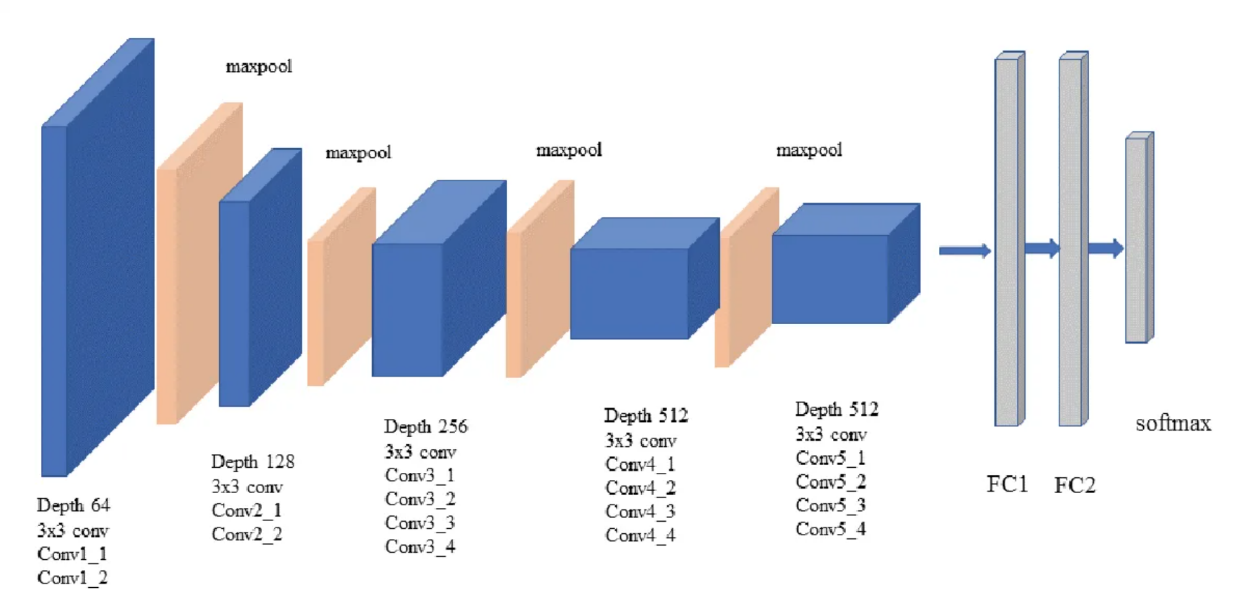


Рисунок 1 – Архитектура VGG19

Разбиение сети на слои имеет принципиальное значение для модели переноса стиля. Как видно на изображении, нейронная сеть VGG19 представляет собой последовательность сверточных блоков (синие прямоугольники), разделенных слоями пулинга, которые уменьшают пространственное разрешение. Оранжевые прямоугольники представляют собой активационные функции ReLU. Начальные слои сети (conv1\_1, conv1\_2) захватывают низкоуровневые детали изображения — края, текстуры и цвета. Средние слои (примерно conv2\_1 - conv4\_2) определяют более сложные паттерны и текстурные элементы. Глубокие слои (conv5\_1 - conv5\_4) абстрагируются до уровня объектов и их композиции.

Программную реализацию считывания слоёв можно увидеть на рисунке 2.8.

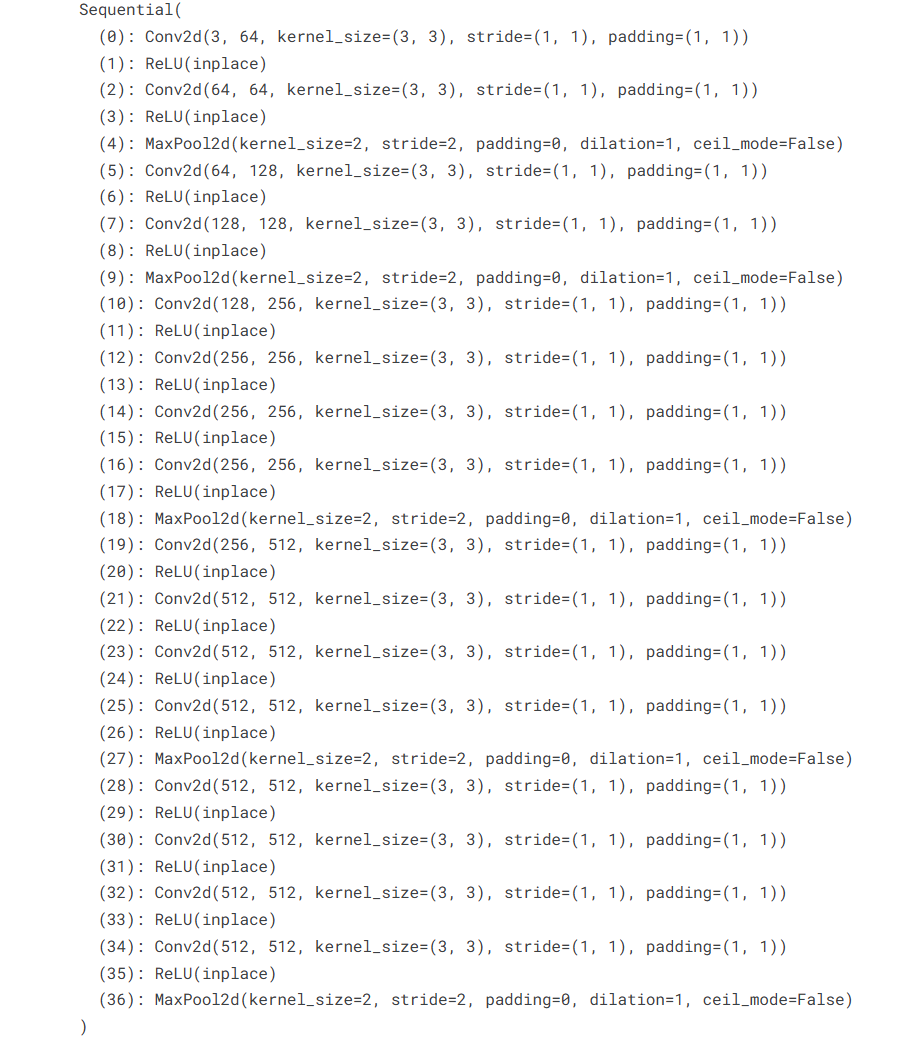


Рисунок 2 – Работа со слоями изображений

Ключевое наблюдение, сделанное исследователями, заключалось в том, что для захвата стиля и содержания необходимо использовать активации разных слоев сети.

Механизм работы модели основан на оптимизации трех ключевых компонентов: контентного изображения (например, фотографии), стилевого изображения (например, картины художника) и выходного изображения, которое изначально может представлять собой случайный шум или копию контентного изображения. Процесс оптимизации направлен на минимизацию двух типов потерь: потери содержания и потери стиля.

Потеря содержания измеряет различие между активациями определенных слоев нейронной сети для контентного и выходного изображений. Эта метрика гарантирует, что генерируемое изображение сохраняет основные структурные элементы и объекты оригинального контентного изображения. Для расчета этой потери обычно используются активации более глубоких слоев сети, которые отвечают за восприятие высокоуровневых характеристик изображения.

Потеря стиля представляет собой более сложную концепцию и является главной инновацией данного подхода. Для захвата стилистических особенностей используется так называемая матрица Грэма. Эта матрица рассчитывается как произведение матрицы активаций определенного слоя на её транспонированную версию. Такой математический прием позволяет определить корреляции между различными каналами активаций, что эффективно отражает текстурные и стилистические характеристики изображения.

Важным аспектом модели является соотношение весов для различных компонентов потери. В таблице ниже показаны типичные значения параметров, используемых при оптимизации:

Таблица 2.4 – Значения параметров при оптимизации

|  |  |  |
| --- | --- | --- |
| Параметр | Типичное значение | Влияние при увеличении |
| Вес содержания (α) | 1 | Более точное сохранение структуры контентного изображения |
| Вес стиля (β) | 1000 | Более сильное проявление стилистических особенностей |
| Вес регуляризации | 0.001 | Более гладкие результаты с меньшим количеством артефактов |

Общая функция потерь представляет собой взвешенную сумму потерь содержания и стиля. Коэффициенты, используемые для взвешивания этих компонентов, позволяют контролировать баланс между сохранением оригинального содержания и степенью стилизации. Если увеличить вес потери содержания, результирующее изображение будет ближе к оригинальной фотографии с легкими стилистическими изменениями. Наоборот, увеличение веса потери стиля приведет к более яркому выражению художественных особенностей стилевого изображения, потенциально за счет четкости и узнаваемости объектов.

Для оптимизации общей функции потерь используется метод градиентного спуска, который постепенно изменяет пиксели выходного изображения, чтобы минимизировать общую потерю. Важно отметить, что в данном подходе обучению подвергается не сама нейронная сеть, а только выходное изображение. Веса предварительно обученной сети остаются фиксированными на протяжении всего процесса оптимизации.

Модель переноса стиля обладает рядом преимуществ по сравнению с другими подходами к генерации художественных изображений. Она предоставляет высокую степень контроля над балансом стиля и содержания, не требует обучения на больших наборах данных и позволяет использовать любые стилевые изображения. Однако у нее есть и определенные ограничения: относительно медленная генерация изображений, зависимость качества от выбора гиперпараметров и возможность создания артефактов в сложных сценах.

### 1.4 Модель style transfer

Технология переноса стиля основывается на уникальной способности глубоких нейронных сетей извлекать и разделять содержание и стилистические особенности изображений. В процессе участвуют три ключевых изображения: контентное, которое определяет, что именно будет изображено на финальной картине; стилевое, задающее художественные особенности; и целевое, которое постепенно трансформируется в ходе работы алгоритма. Ниже опишем ключевые этапы процесса.

Извлечение контентных признаков. В архитектуре VGG19 слой conv4\_2 используется для захвата содержания изображения, так как он сохраняет достаточно высокоуровневые особенности, но не теряет пространственную информацию. Контентная функция потерь определяется как , где – активации 1-го слоя для генерируемого изображения, – активации 1-го слоя для контентного изображения l – слой conv4\_2.

Извлечение стилевых признаков. Стилевые особенности извлекаются из нескольких слоев сети:

* conv1\_1 (низкоуровневые текстуры)
* conv2\_1 (простые формы)
* conv3\_1 (более сложные паттерны)
* conv4\_1 (комплексные особенности)
* conv5\_1 (высокоуровневые абстракции)

Вычисление матриц Грамма для стилевых особенностей. Для каждого слоя l вычисляется матрица Грама , где - k-я активация i-го фильтра в слое l, - k-я активация j-го фильтра в слое l

На рисунке \*\* представлен простой пример работы с матрицей Грама.

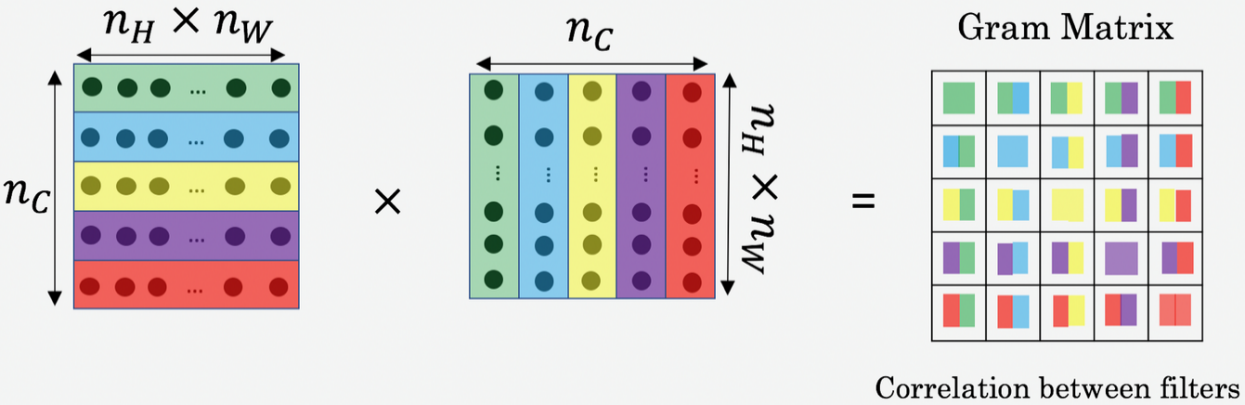


Рисунок 2.9 – матрица Грама

Формула - - демонстрирует, как вычисляется функция потерь стиля для каждого слоя на основе разницы между матрицами Грама, где – количество фильтров в слое l, – размер карты признаков.

Оптимизация с помощью Adam. Общая функция потерь , где – вес контентной составляющей, – вес стилевой составляющей =

Особую роль в процессе играет извлечение признаков изображений. Для сохранения содержания используется слой conv4\_2 сети VGG19, который способен уловить основные структурные элементы изображения, сохраняя при этом достаточную абстрактность представления.

Для передачи стиля задействуется целый каскад слоев, начиная от conv1\_1 и заканчивая conv5\_1, что позволяет захватить стилистические особенности на разных уровнях детализации.

## 2 Программная реализация

В рамках настоящего раздела представлено описание архитектурного подхода к разработке программной системы, включающее формализованный алгоритм взаимодействия с сервисом, обоснование выбора стека и детализацию компонентов.

### 2.1 Общий алгоритм работы приложения

Общий алгоритм взаимодействия пользователем с сервисом “Помогатор” представлен на рисунке 3.1.

Изображение выглядит как текст, диаграмма, снимок экрана, План

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.1 – Общий алгоритм взаимодействия сервиса “Помогатор” и пользователя

### 2.2. Алгоритм применения обученных моделей

Алгоритм применения обученных моделей состоит из следующей последовательности действий:

* Загрузка данных на сервер;
* Предобработка данных;
* Конвертация данных в байты;
* Отправка данных в Kafka топик;
* Получение данных из топика Kafka определенным микросервисом;
* Обработка данных с помощью нейросети или библиотеке в фоновой задаче;
* При успешном выполнении отправляется все в Kafka топик с постфиксом “result”;
* Отправка данных пользователю



### 2.3 Выбор и обоснование инструментальных средств

В основе проекта лежит Python 3.11, выбранный за его читаемость, богатую экосистему и поддержку асинхронного программирования. Для создания микросервисов используется FastAPI, который обеспечивает достаточно высокую производительность и автоматическую генерацию документации благодаря интеграции с OpenAPI. Асинхронная обработка длительных задач, таких как обработка изображений, реализована с помощью Taskiq в связке с Redis, где Redis выступает брокером сообщений, гарантируя надежное распределение задач между воркерами.

Для межсервисного взаимодействия через шину событий выбрана технология - Apache Kafka, которая интегрируется с помощью FastStream, что позволяет писать consumer/producer-логику в декларативном стиле с использованием декораторов и автоматической валидации сообщений через Pydantic. Это обеспечивает гибкость и отказоустойчивость при обработке распределенных транзакций [3].

Для управления зависимостями и качеством кода были выбраны такие технологии, такие как poetry, pre-commit, ruff и mypy, которые позволяют поддерживать чистоту кода, автоматизировать тестирование и гарантировать типизацию с помощью статического анализа.

Для реализации инъекции зависимостей и IoC используется фреймворк dishka, который упрощает управление объектами и их отношениями в разных слоях приложения [4].

Мониторинг состояния системы реализован через prometheus-client для сбора метрик и Grafana для их визуализации, а ошибки отслеживаются с помощью Sentry, что позволяет оперативно реагировать на нештатные ситуации.

Для хранения данных выбрана PostgreSQL, сочетающая надежность реляционной модели с поддержкой JSON и транзакций. Доступ к БД организован через SQLAlchemy с асинхронным драйвером asyncpg.

Безопасность авторизации и аутентификации обеспечивается библиотекой AuthX, использующей JWT.

Работа с изображениями и нейросетями реализована с помощью OpenCV для предобработки данных, NumPy для операций с массивами и TensorFlow для запуска собственных моделей.

Для контейнеризации сервисов применяются Docker и Docker Compose, что гарантирует детерминированность окружения на всех этапах разработки и эксплуатации.

### 2.4 **Программная реализация**

Для реализации проекта была выбрана микросервисная архитектура, основанная на принципах Domain-Driven Design (DDD). Такой подход позволил выделить логически независимые домены в отдельные микросервисы, каждый из которых имеет: собственную бизнес-логику, четко определенные границы (bounded context), возможность автономного развития и масштабирования.

Микросервисы взаимодействуют между собой через асинхронную коммуникацию с использованием шины событий, что обеспечивает гибкость, отказоустойчивость и высокую пропускную способность.

Каждый микросервис построен по принципам DDD и содержит следующие элементы:

* + Entity – объект с уникальным идентификатором;
  + Value Object – неизменяемый объект, представляющий данные;
  + Repository – абстракция доступа к данным внутри домена;
  + Service – содержит бизнес-логику, не связанную напрямую с одной сущностью;
  + Event – события, происходящие в рамках домена и используемые для межсервисной коммуникации;
  + Unit of work – абстракция для управления локальными транзакциями;
  + Bootstrap – механизм для собственной инъекции зависимостей;
  + Message bus – шина сообщений, с помощью которой можно передавать данные.

Внутри каждого микросервиса применяется паттерн CQRS, при котором операции изменения состояния – Commands – и чтения – Views – разделены:

* + Command – запускает действие на изменение данных.
  + View – чтение данных. [1]

Для согласования распределённых транзакций между микросервисами используется паттерн Saga, реализованный через хореографию, без центрального оркестратора. [2].

Приведем теперь практическую реализацию нашего кода в виде UML диграмм. В большинстве случаев иерархия классов совпадает в большинстве микросервисов, поэтому ниже будет представлена иерархия user-service.

На рисунке 3.1 представлена UML диаграмма, которая описывают иерархию событий – Events.

Изображение выглядит как диаграмма, линия, зарисовка, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.1 – UML диаграмма Event классов

Теперь перейдем к рассмотрению UML для классов команд и перехватчиков, включая шину сообщений. Результат представлен на рисунке 3.2.

Изображение выглядит как диаграмма, линия, План, текст

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.2 – UML диаграмма перехватчиков и команд

Ниже на рисунке 3.3 представлена UML диаграмма для классов ошибок. Каждые ошибки были вынесены в мои собственноручные классы для удобства дальнейшего использования в качестве анализа ошибок.

Изображение выглядит как линия, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.3 – UML диаграмма классов ошибок

Ниже на рисунке 3.4 представлена UML диаграмма для классов настроек.

Изображение выглядит как диаграмма, зарисовка, План, Технический чертеж

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.4 – UML диаграмма классов настроек

Теперь перейдем к UML диаграмме классов – репозиториев, которые служат абстракцией для управления последовательностью элементов. В данном случае репозитории могут работать не только с элементами баз данных, но и кэша.

Изображение выглядит как диаграмма, линия, текст, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.5 – UML диаграмма репозиториев

Последняя диаграмма – это отношение доменных сущностей к моему коду. Она представлена на рисунке 3.6.

Изображение выглядит как диаграмма, текст, линия, План

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.6 – отношение доменных моделей одного микросервиса

### 2.5. Демонстрация работы приложения

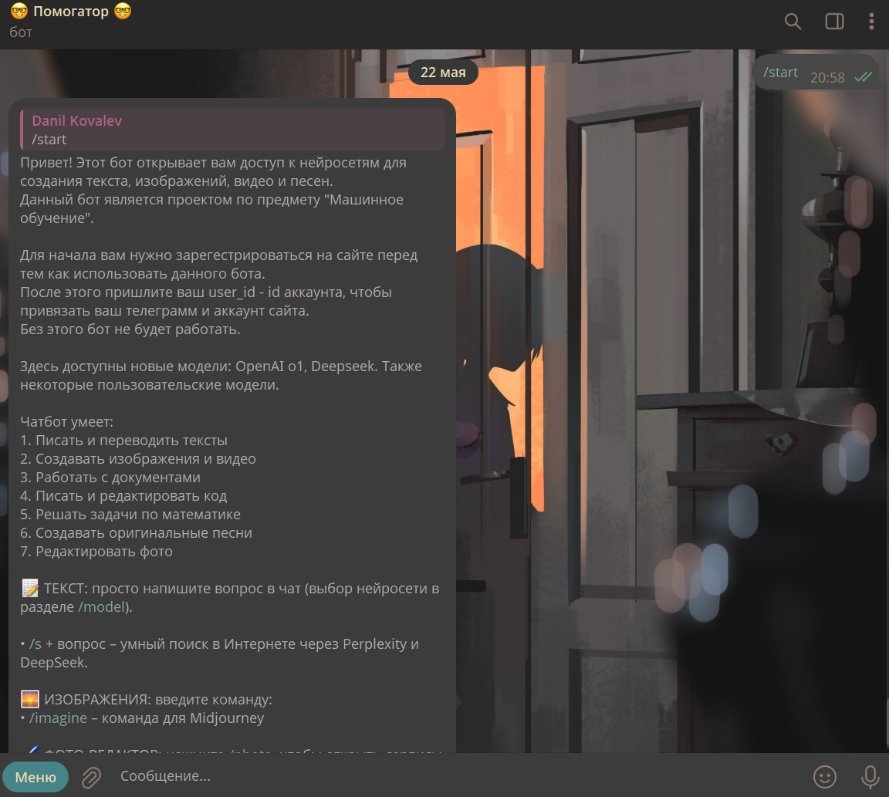


Рисунок 4.6 – стартовое сообщение при удачной аутентификации

После этого пользователь может выбрать различные режимы работы бота. Например, режим художника представлен на рисунке 4.7. Результат обработки представлен на рисунке 4.8.

Изображение выглядит как аниме, Человеческое лицо, мультфильм, одежда

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.7 – прием изображения от пользователя

Изображение выглядит как текст, снимок экрана, Компьютерная игра, мультфильм

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4.8 – результат перевода изображения из серого в цветное

На данный момент сервис “Помогатор” поддерживает также и текстовый режим, в котором пользователь может общаться с DeepSeek, ChatGPT. Работа представлена на рисунке 4.9.

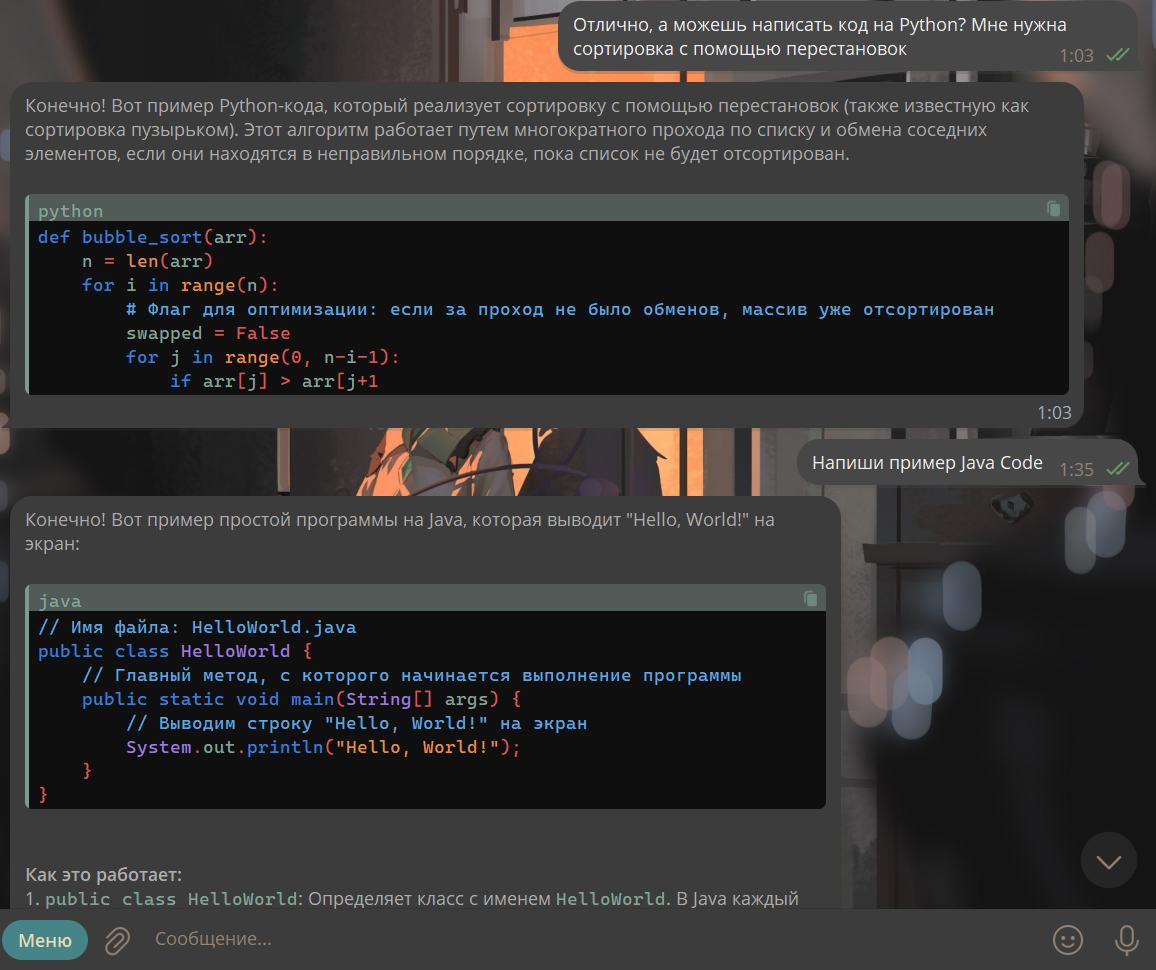


Рисунок 4.9 – обработка текстовых запросов от пользователей

Для мониторинга приложения используется Grafana Stack. Ниже на рисунке 4.10 представлен Grafana dashboard, который показывает нагрузку микросервиса изображений. Под каждый микросервис был сделан собственный dashboard.

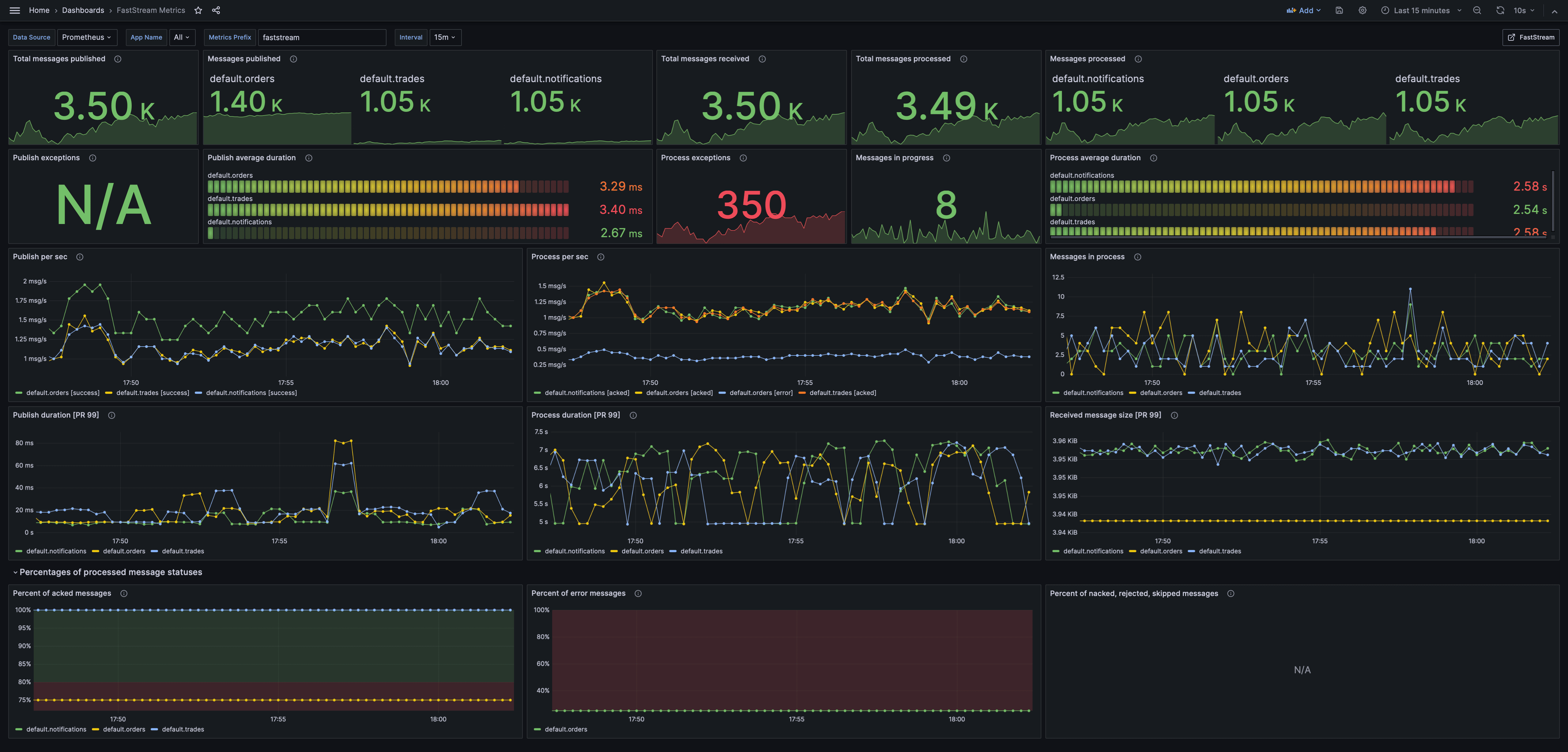


Рисунок 4.10 – Grafana dashboard

Также использовались и внешние инструменты для мониторнига кода. В рамках проекта выбор пал на Sentry. Его интерфейс представлен на рисунке 4.11.

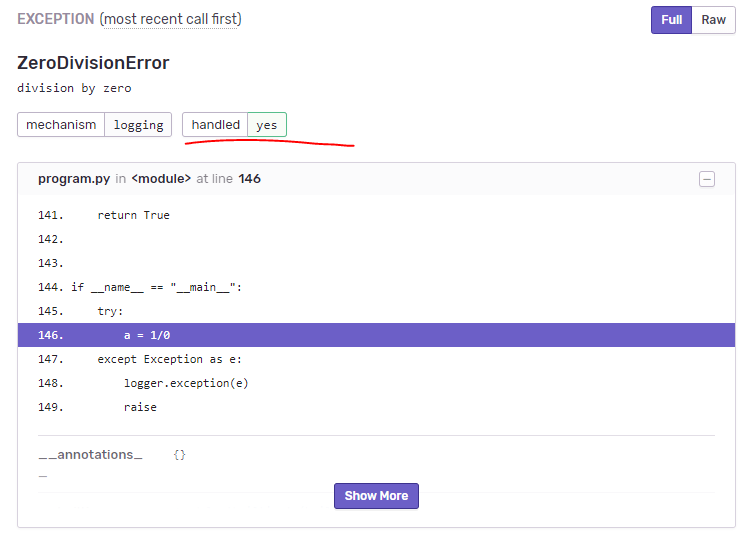


Рисунок 4.11 – интерфейс Sentry

На рисунке 4.12 представлен интерфейс Prometheus, который говорит об успешности подключения к различным микросервисам.

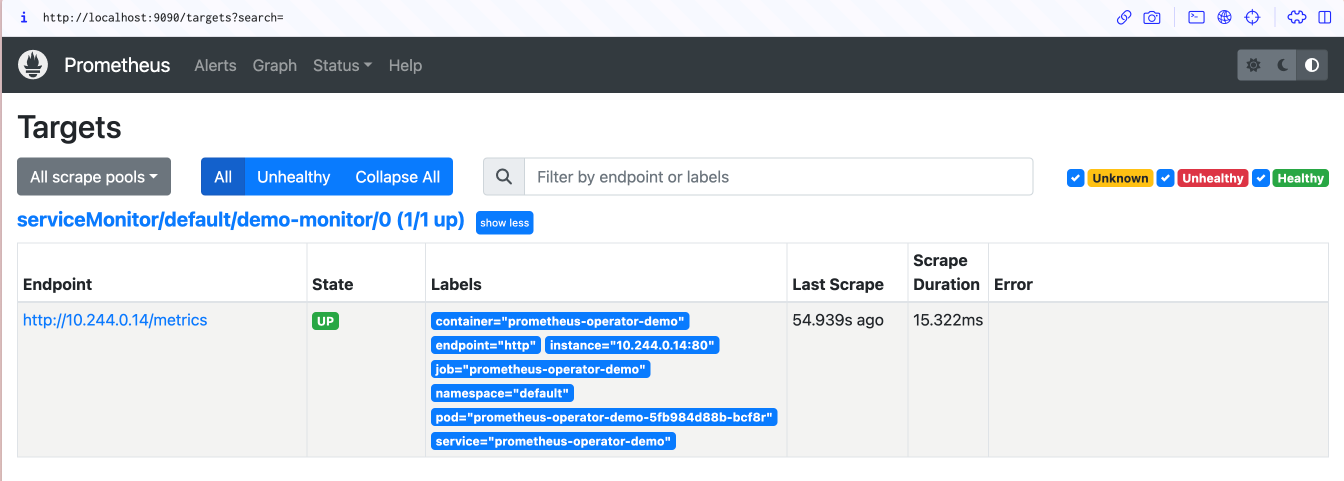


Рисунок 4.12 – интерфейс Prometheus

На рисунке 4.13 представлена документация Swagger для работы с микросервисом пользователей.

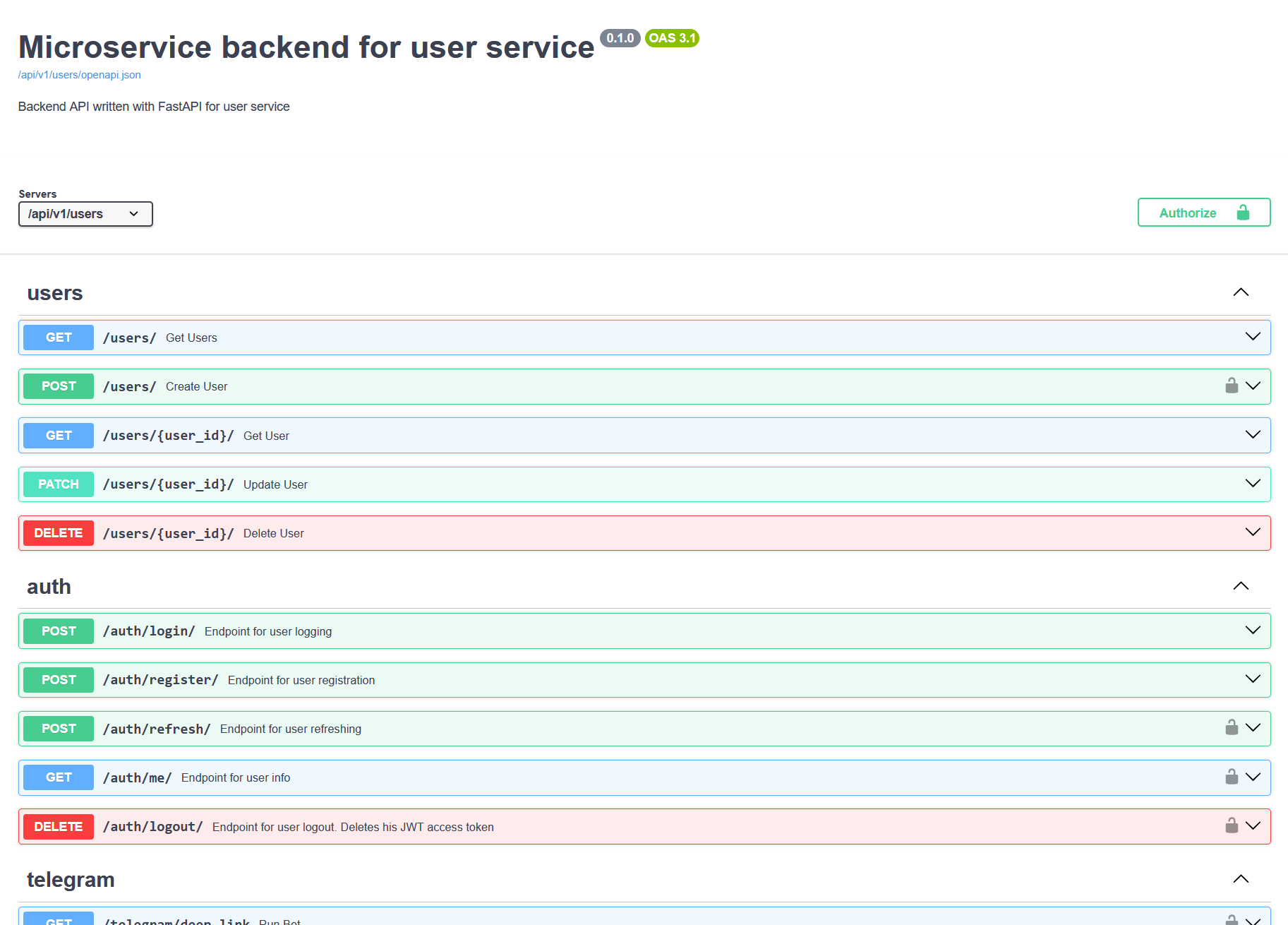


Рисунок 4.13 – документация Swagger для микросервиса пользователей

# Заключение

В рамках выполнения практической подготовки разработана программная система, реализующая автоматизированную обработку данных с использованием технологий искусственного интеллекта и микросервисной архитектуры. Основной задачей проекта стало создание Telegram-бота «Помогатор» — многофункционального инструмента, интегрирующего модели машинного обучения (включая генеративные нейросети и библиотеки компьютерного зрения, такие как OpenCV) для решения задач в сферах дизайна, OSINT-анализа и маркетинга. Разработанное решение полностью соответствует установленным требованиям и демонстрирует высокую степень функциональной завершённости.

Ключевым этапом работы стало формирование обучающего датасета для задачи колоризации черно-белых изображений, включающего 7129 пар изображений (цветное/серое) с предварительно нормализованными характеристиками. Для решения задачи был реализован автоэнкодер на основе свёрточной нейронной сети (CNN), показавший среднюю точность 82–85% на тестовой выборке. Наиболее высокие результаты (90–95%) достигнуты при обработке типичных природных сцен, таких как горные пейзажи и озера, тогда как сложные условия освещения (например, закаты) снизили точность до 70–75%.

Архитектурная реализация системы основана на принципах микросервисности и предметно-ориентированного проектирования (DDD). Взаимодействие между компонентами обеспечивается через распределённую шину событий Apache Kafka, что позволяет достичь масштабируемости, отказоустойчивости и гибкости в управлении вычислительными ресурсами. Асинхронная обработка задач реализована с использованием фреймворка FastStream и брокера Redis, что минимизирует задержки при работе с ресурсоёмкими операциями (например, обработка изображений).

# Перечень использованных информационных ресурсов

1. **Гатченко Н. А, Исаев А. С, Яковлев А.Д.** Криптографическая защита информации. — Санкт-Петербург : НИУ ИТМО 2012. – 142 с. books.ifmo.ru : сайт. - Криптографическая защита информации. – URL: https://books.ifmo.ru/file/pdf/929.pdf (дата обращения 18.06.2024) – Текст: непосредственный.
2. **Рыжков Г., Волкова И. Б.** RSA – алгоритм : исследовательский проект. – Хабаровск : Академия педагогических проектов Российской Федерации – 9 с. педпроект.рф : сайт. - RSA – алгоритм. – URL: https://педпроект.рф/edu-05-2024-pb-154747/ (дата обращения 18.06.2024). – Текст : электронный.
3. SecurityLab.ru: Информационный портал по безопасности : сайт. –Квантовые компьютеры могут в любой момент взломать RSA-шифрование. –URL: https://www.securitylab.ru/news/535527.php (дата обращения 18.06.2024). – Текст: электронный.
4. **Сущевский Д. Г, Панченко О. В, Кугураков В. Н.** Cовременные криптосистемы и их особенности. – Казань : Вестник Казанского технологического университета. – 4 c. kpfu.ru : сайт. - Cовременные криптосистемы и их особенности. – URL: https://repository.kpfu.ru/?p\_id=125683 (дата обращения 18.06.2024). – Текст : электронный.
5. Encyclopedia.kaspersky.ru: База знаний и глоссарий : сайт. – Симметричное шифрование. – URL: https://encyclopedia.kaspersky.ru/glossary/symmetric-encryption/ (дата обращения 18.06.2024). – Текст : электронный.
6. Encyclopedia.kaspersky.ru: База знаний и глоссарий : сайт. – Асимметричное шифрование. – URL: https://encyclopedia.kaspersky.ru/glossary/asymmetric-encryption/ (дата обращения 18.06.2024). – Текст : электронный.

# Приложение А

**Техническое задание на программное средство**

|  |  |
| --- | --- |
| «СОГЛАСОВАННО» | «УТВЕРЖДЕНО» |
| Руководитель проекта | Зав. кафедрой «КБИС» |
| / Д. П. Ковалев / | \_\_\_\_\_\_\_\_\_/ О.А. Сафарьян / |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г. | «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г. |

**А.1 Введение**

Наименование разрабатываемого программного средства: «Программная реализация криптографического алгоритма RSA»

Область применения:

* лабораторные работы по предмету алгоритмы и структуры данных,

криптография;

* личное пользование;
* защита данных.

**А.2 Основания для разработки**

Разработка ведётся на основании документа «Положение о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования», утверждённого министерством образования и науки Российской Федерации 4.05.2018.

Предметным основанием является задание на учебную практику.

**А.3 Назначение разработки**

Функциональное назначение программного средства заключается  
в шифровании и дешифровании сообщения, дешифрованного или же зашифрованного алгоритмом RSA. Изучение криптографического алгоритма RSA для решения различных прикладных задач в программировании.

Эксплуатационное назначение программного средства заключается в проведении лабораторных занятий по информационной безопасности в вузе.

**А.4 Требование к программе или программному изделию**

**А.4.1 Требование к функциональным характеристикам**

Программа корректно использует алгоритмы шифрования и дешифрования RSA

Состав выполняемых функций:

* работать под управлением ОС Windows 7, Windows 8/8.1, а также

Windows 10;

* шифрование и расшифровывание алгоритмом RSA.

Входные данные:

* сообщение и длина ключа.

Выходные данные:

* зашифрованный и расшифрованный текст;

Данные программой хранятся до следующего использование программы или выхода из нее.

**А.4.2 Требование надежности**

Разрабатываемая программа должна удовлетворять следующим требованиям к надежности: отказы программы вследствие внутренних ошибок не допустим;

**А.4.3 Условия эксплуатации**

Для функционирования программного продукта необходимо соблюдение всех требований и правил эксплуатации компьютерной техники. Программное средство обслуживается одним пользователем.

**А.4.4 Требование к составу и параметрам технических средств**

Состав технических средств:

* 5 Гб свободного места на диске; 32 или 64-разрядный процессор с

частотой не ниже 1 ГГц;

* монитор VGA или HDMI;
* клавиатура и мышь;
* оперативная память не меньше 16 Гб;
* интерпретатор Python 3.12 требуется;
* система виртуализации Docker.

**А.4.5 Требования к информационно и программной совместимости**

Программное средство требует семейство операционных систем GNU/Linux для взаимодействия с пользователем через терминал.

**А.4.6 Требования к маркировке и упаковке**

Требования к маркировке не предъявляются. Требования к упаковке определяются требованиями к хранению носителей информации, содержащих ПО.

**А.4.7 Требования к транспортировке и хранению**

Требования к транспортировке и хранению определяются требованиями к транспортировке и хранению носителей информации, содержащих ПО.

**А.6 Стадии и этапы разработки**

Ознакомление с теоретическим материалом (от 10.06.25 до 15.06.25);

Оформление программной реализации (от 16.06.25 до 29.06.25);

Испытание программного средства (от 29.06.25 до 3.07.25).

**А.7 Порядок и контроль приемки**

Порядок и контроль приемки определяется руководством кафедры КБИС Донского государственного технического университета.

Разработчик технического задания: Ковалев Данил Петрович

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2025г \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

# Приложение Б

**Листинг 1**

Б.1 – Реализация «main.py»:

import logging

from collections.abc import AsyncGenerator

from contextlib import asynccontextmanager

from typing import Final

from dishka import AsyncContainer

from dishka.integrations.fastapi import setup\_dishka as setup\_dishka\_fastapi

from dishka.integrations.faststream import setup\_dishka as setup\_dishka\_faststream

from fastapi import FastAPI

from fastapi.middleware.cors import CORSMiddleware

from faststream import FastStream

from faststream.asgi import make\_ping\_asgi, make\_asyncapi\_asgi

from faststream.kafka import KafkaBroker

from sqlalchemy.ext.asyncio import AsyncEngine, create\_async\_engine

from sqlalchemy.orm import clear\_mappers

from starlette.middleware.trustedhost import TrustedHostMiddleware

from app.application.api.middlewares.metrics import HTTPLatencyMetricsMiddleware

from app.application.api.v1.auth.handlers import router as auth\_router

from app.application.api.v1.users.handlers import router as users\_router

from app.application.api.v1.utils.handlers import register\_exception\_handlers

from app.infrastructure.adapters.alchemy.orm import metadata, start\_mappers

from app.infrastructure.brokers.base import BaseMessageBroker

from app.logic.container import get\_container

from app.settings.config import Settings, get\_settings

from app.settings.logger.config import setup\_logging

from app.application.api.v1.telegram.handlers import router as telegram\_router

logger: Final[logging.Logger] = logging.getLogger(\_\_name\_\_)

@asynccontextmanager

async def lifespan(app: FastAPI) -> AsyncGenerator[None, None]:

settings: Settings = get\_settings()

container: AsyncContainer = get\_container()

broker: BaseMessageBroker = await container.get(BaseMessageBroker)

faststream\_broker: KafkaBroker = await container.get(KafkaBroker)

faststream\_app: FastStream = FastStream(faststream\_broker, logger=logger)

setup\_dishka\_faststream(

container, FastStream(faststream\_broker, logger=logger), auto\_inject=True

)

# cache.pool = await container.get(ConnectionPool)

# cache.client = await container.get(Redis)

engine: AsyncEngine = create\_async\_engine(settings.database.url)

async with engine.begin() as conn:

await conn.run\_sync(metadata.create\_all)

start\_mappers()

app.mount("/health", make\_ping\_asgi(faststream\_broker, timeout=5.0))

app.mount("/asyncapi", make\_asyncapi\_asgi(faststream\_app))

await broker.start()

yield

await broker.stop()

await app.state.dishka\_container.close()

clear\_mappers()

def create\_app() -> FastAPI:

settings: Settings = get\_settings()

container: AsyncContainer = get\_container()

app = FastAPI(

title="Microservice backend for user service",

description="Backend API written with FastAPI for user service",

debug=True,

root\_path="/api/v1/users",

lifespan=lifespan,

)

setup\_logging()

register\_exception\_handlers(app)

app.add\_middleware(

CORSMiddleware, # type: ignore

allow\_origins=settings.cors.allow\_origins, # Разрешить все домены

allow\_credentials=settings.cors.allow\_credentials,

allow\_methods=settings.cors.allow\_methods,

allow\_headers=settings.cors.allow\_headers,

)

app.add\_middleware(

HTTPLatencyMetricsMiddleware, container=container # type: ignore

)

app.add\_middleware(

TrustedHostMiddleware, # type: ignore

allowed\_hosts=settings.server.allowed\_hosts,

)

app.include\_router(users\_router)

app.include\_router(auth\_router)

app.include\_router(telegram\_router)

setup\_dishka\_fastapi(container=container, app=app)

return app

import cv2

import numpy as np

from app.domain.entities.image import ImageEntity

from app.exceptions.infrastructure import Cv2ImageDecodingError

from app.infrastructure.integrations.color\_to\_gray.base import (

BaseImageColorToCrayScaleConverter,

)

class Cv2ImageColorToCrayScaleConverter(BaseImageColorToCrayScaleConverter):

@override

def convert(self, image: ImageEntity) -> ImageEntity:

cv2\_image: cv2.typing.MatLike = cv2.imdecode(

np.frombuffer(image.data, dtype=np.uint8), cv2.IMREAD\_COLOR

)

if cv2\_image is None:

raise Cv2ImageDecodingError("Failed to decoding image")

gray\_image = cv2.cvtColor(cv2\_image, cv2.COLOR\_BGR2GRAY)

\_, encoded\_img = cv2.imencode(".jpg", gray\_image)

return ImageEntity(

data=encoded\_img.tobytes(),

width=image.width,

height=image.height,

name=image.name,

)

from abc import ABC, abstractmethod

from app.domain.entities.image import ImageEntity

class BaseImageColorToCrayScaleConverter(ABC):

@abstractmethod

def convert(self, image: ImageEntity) -> ImageEntity:

raise NotImplementedError

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *№ строки* | *Формат* | *Обозначения* | | | | *Наименование* | | | | *Кол-во листов* | | *Примечание* | |
|  |  |  | | | | *Документация общая* | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
| *1* | *А4* |  | | | | *Задание на практику* | | | | *1* | |  | |
| *2* | *А4* | *УП.990000.000* | | | | *Отчет по учебной практике (включает 14 рисунков)* | | | | *29\** | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | | *Документация приложений* | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
| *3* | *А4* | *УП.990000.000* | | | | *Приложение А-Техническое задание на программное средство* | | | | *4* | |  | |
| *4* | *А4* | *УП.990000.000* | | | | *Приложение Б-Листинг кода* | | | | *8* | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  | | | |  | | | |  | |  | |
|  |  |  |  |  | *УП.990000.000* | | | | | | | | |
|  |  |  |  |  |
| *Изм* | *Лист* | *№ Докум.* | *Подп.* | *Дата* |
| *Разраб.* | | *Ковалев Д. П.* |  |  | *Отчет по практической подготовке при проведении учебной практики (ведомость отчета)* | | *Лит.* | | | | *Лист* | | *Листов* |
| *Руков.пр* | | *Бурякова*  *О. С.* |  |  |  |  |  | | *42* | | *1* |
|  | |  |  |  | *ДГТУ*  *Группа ВКБ32* | | | | | | |
| Н. Контр | |  |  |  |
| *Утверд.* | | *Сафарьян О. А.* |  |  |

\*количество листов отчета, исключая приложения