Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**РАЗРАБОТКА GAN нейросети**

Пояснительная записка к курсовому проекту

ПМ.02 Осуществление интеграции программных модулей

МДК.02.01 Технология разработки программного обеспечения

НАТКиГ.202000.010.000ПЗ

Разработал:

студент группы ПР-21.106

Шестаков И.В.

2023

**Содержание**

[Введение 3](#_Toc117112970)

[1 Описание предметной области 3](#_Toc117112971)

[2 Выбор технологии, языка и среды программирования 11](#_Toc117112976)

[3 Анализ и уточнение требований к программногу продукту 17](#_Toc117112979)

[3.1 Выбор методов и разработка основных алгоритмов решения задачи 17](#_Toc117112980)

[4 Разработка структурной схемы программного продукта 30](#_Toc117112983)

[4.1 Описание используемых процедур и библиотечных функций 30](#_Toc117112984)

[4.2 Спецификация программы 34](#_Toc117112985)

[5 Проектирование интерфейса пользователя 30](#_Toc117112983)

[5.1 1Разработка форм ввода-вывода информации 30](#_Toc117112984)

[Заключение 39](#_Toc117112986)

[Литература 40](#_Toc117112987)

[Приложение 1 Техническое задание 41](#_Toc117112988)

Введение

Нейронные сети вдохновлены биологической нейронной системой человека. Нейронные сети состоят из искусственных нейронов, которые объединяются в слои и обрабатывают данные, выполняя различные вычислительные задачи. Они используются для решения широкого спектра задач, включая распознавание образов, классификацию, регрессию, генерацию данных и многое другое.

Генеративно-состязательные сети в последние годы привлекают пристальное внимание исследователей в области машинного обучения и компьютерного зрения. Успехи в разработке GAN открывают путь к созданию более "интеллектуальных" систем, способных решать сложные прикладные задачи. Поэтому развитие GAN является важным направлением в развитии искусственного интеллекта.

Целью курсового проекта является создания генерирующей изображения нейросети.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение проекта с точки зрения пользователя для выявления функций приложения;
* написание кода нейросети;
* тестирование полученного продукта.

Объект исследования – генерирующая нейросеть.

# Описание предметной области

Генеративно-состязательные сети (GAN) — это класс нейросетевых архитектур, используемых для генерации синтетических изображений. В GAN состязаются друг с другом две нейронные сети: генератор и дискриминатор.

Генератор — это нейросеть, которая генерирует новые изображения на основе случайного входного вектора шума. Ее задача - создавать реалистичные изображения, которые нельзя отличить от настоящих.

Дискриминатор — это нейросеть, которая принимает на вход как настоящие, так и сгенерированные изображения. Ее задача - определить, является ли изображение настоящим или сгенерированным.

Пользователь, генерирующий изображение может:

1. Ввод данных:

* Подготовка входных данных (описание желаемого результата);
* Уточнение положительных и отрицательных запросов.

1. Настройка параметров генерации:

* Выбор размера генерируемого изображения;
* Задание количества генерируемых изображений;
* Настройка дополнительных параметров генератора и дискриминатора.

1. Запуск генерации:

* Отправка входных данных в модель GAN;
* Получение на выходе сгенерированных нейросетью изображений.

В процессе обучения GAN дискриминатор учится все лучше отличать настоящие изображения от сгенерированных, а генератор совершенствуется в генерации все более реалистичных изображений, чтобы одурачить дискриминатор.

Процесс обучения GAN, может быть сложным и требовать значительных вычислительных ресурсов. Также GAN могут страдать от проблем, как коллапс моды или нестабильность обучения. Поэтому при реализации GAN необходимо тщательно настраивать гиперпараметры и архитектуру сетей.

Процесс обучения нейросети, можно разделить на несколько ключевых этапов:

1. Подготовка данных:

* Собираются и подготавливаются данные, которые будут использованы для обучения нейросети.

1. Создание генератора и дискриминатора:

* Определяются архитектуры генератора и дискриминатора.

1. Определение функций потерь:

* Выбираются функции потерь для обеих нейросетей. Для GAN применяют бинарную кросс-энтропию.

1. Обучение:

* Генератор создает синтетические изображения на основе случайных входных данных;
* Дискриминатор оценивает синтетические и реальные изображения, предсказывая, является ли изображение реальным или сгенерированным;
* Генератор и дискриминатор обновляют свои веса на основе результатов.

1. Настройка параметров:

* Подбираются гиперпараметры, такие как скорость обучения, количество слоёв и нейронов в сетях.

Обучение GAN требует большого набора реальных изображений в качестве примеров. Чем больше и разнообразнее тренировочные данные, тем лучше GAN сможет научиться генерировать реалистичные изображения.

GAN могут использоваться для генерации фотографических изображений людей, животных, пейзажей и других объектов. Они позволяют создавать высококачественные изображения, неотличимые от реальных фотографий.

GAN открывают большие возможности для синтеза фотореалистичных изображений, которые могут использоваться в различных областях - от развлечений до науки.

В заключение, генеративно-состязательные сети (GAN) представляют собой захватывающую область искусственного интеллекта, которая вдохновлена биологической нейронной системой человека. Эти сети, состоящие из генератора и дискриминатора, открывают огромные возможности для создания синтетических изображений, неотличимых от настоящих фотографий.

Процесс обучения GAN может быть сложным и требовать внимательной настройки, но результаты могут быть впечатляющими. Они могут использоваться в различных областях, включая искусство, развлечения и научные исследования, что делает их важными для развития технологий и креативности. С развитием архитектур и методов обучения GAN можно ожидать, что эта область будет продолжать вдохновлять исследователей и инженеров, открывая новые горизонты в создании фотореалистичных изображений и других приложениях искусственного интеллекта.

# Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Visual Studio Code, так как она предоставляет легкую и интуитивно понятную среду разработки. Её установка и настройка занимают минимум времени, что важно для быстрого старта проекта. К тому же VS Code обладает отличной поддержкой языка программирования Python. Интеграция с Python расширением обеспечивает возможность работы с виртуальными средами, авто дополнение кода, проверку синтаксиса и быстрое исправление ошибок.

Языком программирования выбран Python, так как он является одним из наиболее популярных языков программирования в области машинного обучения и исследовательских задач. Это обеспечивает доступ к богатому экосистему библиотек и инструментов, специально созданных для работы с данными и моделями машинного обучения. Синтаксис Python является простым и читаемым, что облегчает понимание и сопровождение кода. Это особенно важно в научном и исследовательском программировании, где ясность и понятность кода приобретают ключевое значение.

Python обладает обширной библиотекой для работы с данными, научных вычислений и машинного обучения. Библиотеки, такие как NumPy, Pandas, TensorFlow и Keras, предоставляют мощные инструменты для создания и обучения моделей, а также обработки данных.

Выбор технологий обоснован стремлением использовать мощные фреймворки TensorFlow и Keras для создания и обучения моделей глубокого обучения, интеграцией с предварительно обученными моделями текста TensorFlow Hub и TensorFlow Text, а также применением метода гауссовской диффузии для генерации изображений.

# Анализ и уточнение требований к программному продукту

## Выбор методов и разработка основных алгоритмов решения задачи

GENERATING AI включает в себя реализацию гауссовской диффузии, модели для обработки текста с использованием BERT, а также диффузионной модели для генерации изображений.

Процесс диффузии представляет собой постепенное преобразование структурированного сигнала (например, изображения) в шум. Путем моделирования диффузии можно генерировать зашумленные изображения на основе обучающих изображений и обучать нейронную сеть устранению шума. С использованием обученной сети можно воспроизводить обратный процесс диффузии, то есть создавать изображения из шума.

Класс GaussianDiffusion представляет собой утилиту для реализации гауссовской диффузии. Этот метод используется для генерации шума и диффузии входных данных. Гауссовская диффузия является процессом, который вводит случайные шумы в изображения, постепенно увеличивая их дисперсию с течением времени. Математически, гауссовская диффузия определяется следующим образом.

Для временного шага и начального изображения :

, (1)

где коэффициент диффузии, увеличивающийся с течением времени.

После прямой диффузии начинается обратный процесс, в ходе которого изображение постепенно очищается от шума. Это позволяет восстанавливать исходное изображение.

Для обработки текста используется модель BERT (Bidirectional Encoder Representations from Transformers). Эта модель способна создавать контекстные эмбеддинги для текстовых данных, улавливая их семантические свойства и взаимосвязи между словами. Эмбеддинги текста, полученные с использованием BERT, используются в дальнейшем в процессе генерации изображений. BERT обладает трансформерной архитектурой, которая позволяет модели эффективно обрабатывать последовательности данных, сохраняя контекстуальные зависимости между словами.

**def** **process\_text**(text\_batch):

text\_preprocessed = bert\_preprocess\_model(text\_batch)

bert\_results = bert\_model(text\_preprocessed)

**return** bert\_results["pooled\_output"]

Таким образом, BERT преобразует текстовую информацию в числовое представление, сохраняя контекстуальные и семантические зависимости между словами. Полученные эмбеддинги затем используются в процессе генерации изображений с учетом текстового контекста.

Диффузионная модель представлена классом DiffusionModel. Этот класс использует нейросетевую модель (network), обученную для генерации изображений, и предоставляет методы для генерации изображений с использованием гауссовской диффузии.

total\_timesteps = **500**

gdf\_util = GaussianDiffusion(timesteps=total\_timesteps)

network = tf.keras.models.load\_model(model\_path)

model = DiffusionModel(network=network, ema\_network=network, gdf\_util=gdf\_util, timesteps=total\_timesteps)

\_ = model.plot\_images(num\_rows=**2**, num\_cols=**4**, annotation="Clear room", ex\_rate=**2**)

Приведенный код создает экземпляр диффузионной модели, использует ее для генерации изображений на основе текстового описания ("Clear room") и визуализирует результаты. Экстраполяция шума (ex\_rate) позволяет управлять уровнем вариации в генерируемых изображениях.

Обучение модели происходит в цикле, включающем следующие шаги, реализованные в коде:

* Выборка Данных: извлечение батчей из обучающего набора данных, представляющих изображения и соответствующие текстовые описания;
* Случайная Генерация Временных Шагов: в коде реализована случайная выборка временных шагов для применения гауссовской диффузии к изображениям;
* Обновление Весов EMA Модели: после каждого шага обучения веса модели обновляются в соответствии с бегущим средним (EMA) для стабилизации результатов.

Обучение модели осуществляется минимизацией ошибки между исходным шумом и предсказанным шумом:

, (2)

где ;;*.*

Используя обученную модель, предсказывается зашумленное изображение на основе случайного шума:

, (3)

где ;;

;

.

Обучение модели и выборка из неё можно представить в виде алгоритмов на рисунке 1:

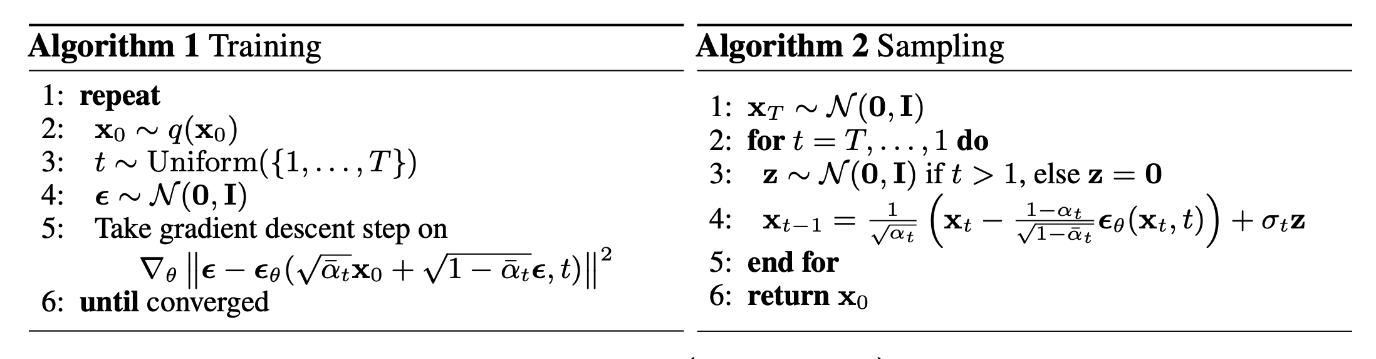


Рисунок 1 – алгоритмы обучения и выборки модели

# Тестирование

## Протокол тестирования дизайна приложения

Тестирование дизайна приложения проводится на самом минимальном (Android SDK 24) и на более позднем (Android SDK 33) с различной диагональю экранов для проверки разметки страниц и вёрстки приложения.

Примеры проверок отображения элементов на экране представлены на рисунках 13–17.

Рисунок 13 – Экраны авторизации

Элементы интерфейса в обоих случаях отображаются корректно.

Рисунок 14 – Экраны авторизации

Экран авторизации так же отображается корректно на обоих устройствах, все элементы интерфейса расположены на своих местах.

Рисунок 15 – Главные экраны

Главный экран тоже отображается корректно. Верхняя и нижняя панель нормальных размеров, элементы на них не сдвинуты. Логотип так же находится на своём месте. Кнопки избранного привязаны к видеопроигрывателю.

Рисунок 16 – Экран добавления видеоролика

На экране добавления видеоролика так же не обнаружено ошибок. Масштаб элементов сохранен. Верхняя панель находится на своём месте и имеет нормальные размеры.

Рисунок 17 – Экраны профиля

Элементы экрана профиля отображаются корректно. Фотография пользователя не поменяла свои масштабы. Элементы меню профиля находятся на своих местах. Текст не растянут и не выходит за пределы экрана.

## Протокол тестирования функционала приложения

Так же необходимо проверить функционал приложения. Для этого, для каждой функции были разработаны TestCase.

В таблице 5 представлено тестирование функции вывода видеоролика на экран.

Таблица 5 – Тестирование функции вывода видео на экран

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Вывод видео на экран | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Окно для вывода видеороликов открылось | пройден |

В таблице 6 представлено тестирование функции добавления видео в «Понравившиеся».

Таблица 6 – Тестирование функции добавление видео в «Понравившиеся».

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Добавление видео в «Понравившиеся» | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Окно для вывода видеороликов открылось | пройден |
| Нажать на кнопку добавления видео в избранное | В базу данных отправлен запрос на добавление видео в избранное | пройден |
| Перейти на вкладку «Понравившиеся» | Добавленный ролик отобразится в списке | пройден |

В таблице 7 представлено тестирование функции загрузки видео в базу данных.

Таблица 7 – Тестирование функции загрузки видео

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Загрузка видео | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Перейти на вкладку добавления видеоролика | Окно корректно загрузилось | пройден |
| Выбрать видеоролик из галереи или снять новый на камеру | Приложение получило файл видеоролика и информацию о нём | пройден |
| Ввести название видеоролика | Приложение получило название видеоролика | пройден |
| Нажать на кнопку «Загрузить» | Файл видеоролика отправлен в базу данных | пройден |
|  |  |  |
|  |  |  |
| **Шаги** **теста (negative)**: |  |  |
| Перейти на вкладку добавления видеоролика | Окно корректно загрузилось | пройден |
| Не выбирать видеоролик | Приложение получит код ошибки | пройден |
| Не вводить название видео | Приложение получит код ошибки | пройден |
| Нажать на кнопку «Загрузить» | Появится всплывающее сообщение с определенной ошибкой | пройден |

В таблице 8 представлено тестирование функции регистрации нового пользователя.

Таблица 8 ­– Тестирование функции регистрации пользователя

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Регистрация нового пользователя | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Перейти на вкладку добавления регистрации | Окно корректно загрузилось | пройден |
| Заполнить обязательные поля корректными данными | Успешное заполнение обязательных полей | пройден |
| Нажать на кнопку «Регистрация» | Приложение проверило введенные данные и зарегистрировало пользователя | пройден |
|  |  |  |
|  |  |  |
| **Шаги** **теста (negative)**: |  |  |
| Перейти на вкладку добавления регистрации | Окно корректно загрузилось | пройден |
| Не заполнять обязательные поля или ввести некорректные данные | Успешное заполнение обязательных полей | пройден |
| Нажать на кнопку «Регистрация» | Появится всплывающее сообщение с определенной ошибкой | пройден |

В таблице 9 представлено тестирование функции авторизации пользователя.

Таблица 9 ­– Тестирование функции авторизации

|  |  |
| --- | --- |
| **Название**: | VideoHost |
| **Функция**: | Авторизация |

Продолжение таблицы 9

|  |  |  |  |
| --- | --- | --- | --- |
| **Действие** | **Ожидаемый** **результат** | | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  | |
|  |  |  | |
| **Шаги** **теста (positive)**: |  | | |
| Заполнить обязательные поля корректными данными | Успешное заполнение обязательных полей | пройден | |
| Нажать на кнопку «Войти» | Приложение проверило введенные данные и авторизировало пользователя | пройден | |
|  |  |  | |
|  |  |  | |
| **Шаги** **теста (negative)**: |  |  | |
| Не заполнять обязательные поля или ввести некорректные данные | Успешное заполнение обязательных полей | пройден | |
| Нажать на кнопку «Войти» | Появится всплывающее сообщение с определенной ошибкой | пройден | |

Разработанные TestCase демонстрируют корректную работу функций приложений. Так же в процессе тестирования не выявлено ошибок в дизайне приложения и его логике. Каждый экран прошёл проверку на разных API на корректное отображение элементов соответствующих экранов.

Заключение

В ходе разработки мобильного приложения для видеохостинга учтены предпочтения и потребности пользователей, в формате коротких видеороликов. Функция ведения списка просмотренного контента успешно реализована, позволяя пользователям отслеживать свои предпочтения и легко вернуться к уже просмотренному контенту.

Интерфейс приложения разработан с учетом удобства использования и минимального количества действий для достижения пользовательской цели. Основные экраны спроектированы таким образом, чтобы не перегружать пользователей лишней информацией, и добавлены полезные функции, такие как просмотр трейлеров и возможность создания собственных списков просмотренного и избранного контента.

Важным элементом концепции приложения является возможность просмотра коротких видеороликов, которая позволяет пользователям сократить время в дороге на работу, учебу или в перерывах между делами. Это удобный способ получать информацию на ходу.

Поддержка списка избранного контента также оказалась значимой для пользователей, предоставляя удобный способ отслеживания и доступа к контенту, который им особенно нравится.

Важным этапом в разработке приложения было тестирование дизайна, которое подтвердило его корректность и успешное функционирование. Это гарантирует, что разработанное приложение соответствует поставленным целям и требованиям.

В целом разработка мобильной версии видеохостинга позволила создать функциональное и удобное приложение, учитывающее потребности пользователей. Проделанная работа успешно достигла поставленной цели курсового проекта и является основой для дальнейшего развития и улучшения приложения.

Библиография

1 Yasno [Электронный ресурс]: Мобильный интерфейс: разработка дизайна приложений и проектирование макетов на примерах – Режим доступа к руководству: <https://yasno.mobi/blog/mobilnyy-interfeys-razrabotka-dizayna-prilozheniy-i-proektirovanie-maketov-na-primerakh/>

2 DevelopersAndroid [Электронный ресурс]: Полная документация по Android–Режим доступа к руководству: <https://developer.android.com/>

3 METANIT [Электронный ресурс]: Документация по языку программирования Kotlin – Режим доступа к руководству: <https://metanit.com/kotlin/>

4 Webkyrs [Электронный ресурс]: Создание базы данных Firebase – Режим доступа к руководству: <https://webkyrs.info/page/kak-sozdat-bazu-dannykh-firebase-realtime-database>

5 Epochta [Электронный ресурс]: как составить портрет клиента – Режим доступа: <https://www.epochta.ru/blog/portret-klientov>

**Приложение А**

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

**РАЗРАБОТКА Мобильной версии видеохостинга**

Техническое задание

НАТКиГ.202000.010.000ПЗ

Выполнил:

Студент группы ПР-21.106

Шестаков И.В.

2023

**Содержание**

Введение 43

1 Назначения разработки 44

2 Требования к мобильному приложению 45

2.1 Требования к функциональным характеристикам 45

2.2 Требования к надёжности 45

2.3 Условия эксплуатации 45

2.4 Требования к составу и параметрам технических средств 45

2.5 Требования к информационной и программной совместимости 46

2.6 Требования к защите информации 46

2.7 Требования к маркировке и упаковке 46

3 Требования к программной документации 47

4 Технико-экономические показатели 48

5 Стадии и этапы разработки 49

6 Порядок контроля и приёмки 50

**Введение**

Настоящее техническое задание распространяется на разработку мобильного приложения «Разработка мобильной версии видеохостинга», используемого для просмотра видеороликов, ведения списков загруженных видеороликов и понравившегося.

Наименование приложения: «VideoHost».

Краткая характеристика области применения: мобильное приложение предоставляет возможность просмотра видеороликов, загруженных на видеохостинг, а также ведения списка загруженных видеороликов самого пользователя и наиболее понравившегося ему контента.

Основанием для проведения разработки является Протокол №6 от 21 февраля 2022 года.

Наименование темы разработки − «Разработка мобильной версии видеохостинга».

Условное обозначение темы разработки – «VideoHost».

**1 Назначение разработки**

Основное назначение приложения заключается в:

* обеспечении удобного и понятного интерфейса приложения для пользователя;
* предоставление большой библиотеки видеороликов.

Лица, которые могут работать с данной системой:

администратор – управляет полностью всем приложением, следит за его работоспособностью, обновляет информацию в базе данных приложения;

пользователь приложения – может просматривать видеоролики загруженные на видеохостинг, а также вести списки понравившихся и загруженных им видеороликов.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* просмотр видеороликов;
* просмотр списка загруженных пользователем видеороликов;
* просмотр списка понравившихся видеороликов;

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация стабильного интернет-соединения.

Приложение должно контролировать входную информацию:

* соблюдение типов данных при заполнении полей.

**2.3 Условия эксплуатации**

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

**2.4 Требования к составу и параметрам технических средств**

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android не ниже версии 7.0.

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Java.

**2.6 Требования к защите информации**

Доступ к информации БД предоставляется только администратору базы данных.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* пояснительная записка.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Этапы разработки КП** | **Сроки выполнения** | **Отчётность** |
| 1 | Определение цели и задач, объекта и предмета исследования | 13.03.2023 | Пояснительная записка |
| 2 | Описание предметной области | 08.04.2023 | Пояснительная записка |
| 3 | Выбор технологии, языка и среды программирования | 10.04.2023 | Пояснительная записка |
| 4 | Оформление технического задания | 15.04.2023 | Техническое задание |
| 5 | Проектирование UI/UX дизайна | 17.04.2023 | Спецификации программного обеспечения |
| 6 | Разработка мобильного приложения | 22.04.2023 | Схема структурная системы и спецификации компонентов |
| 7 | Разработка базы данных | 02.05.2023 | Программный  продукт |
| 8 | Отладка и тестирование приложения | 13.05.2023 | Тексты программных компонентов |
| 9 | Оформление документации | 22.05.2023 | Программная  документация |
| 10 | Защита | 27.05.2023 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт;