**Отчет Практической Работы**

Языки Программирования

Десктопное приложение генерации текста на основе ChatGPT

**Группа:** КИСП-9-22

**Команда:** ОГУЗКИ

Степанов Руслан (КИСП-9-22 (2)

Слепцов Евгений (КИСП-9-22 (2)

Цвикальский Артём (Pre-Alpha) (КИСП-9-22 (2)

Лаптев Никита (Alpha) (КИСП-9-22 (2)

Семёнов Юрий (Pre-Release) (КИСП-9-22 (2)

Преподаватель: Федоров Дьулуур Андрианович

Якутск – 2025

Содержание

[ВВЕДЕНИЕ 3](#_Toc199429196)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc199429197)

[1.1 Терминология по проекту и глоссарий 5](#_Toc199429198)

[1.2 Распределение ролей и работы 7](#_Toc199429199)

[1.3 Стек Технологий 9](#_Toc199429200)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ (РАЗРАБОТКА) 10](#_Toc199429201)

[2.1 Архитектура ПО 10](#_Toc199429202)

[2.2 Разработка Проекта по ролям 11](#_Toc199429203)

[2.3 Контроль выполнения плана 14](#_Toc199429204)

[ЗАКЛЮЧЕНИЕ 15](#_Toc199429205)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc199429206)

[ПРИЛОЖЕНИЕ 1 17](#_Toc199429207)

# ВВЕДЕНИЕ

ChatGPT (Generative Pre-Trained Transformer) — Чат-бот с генеративным искусственным интеллектом, разработанный компанией OpenAI и способный работать в диалоговом режиме, поддерживающий запросы на естественных языках.

Система способна отвечать на вопросы, генерировать тексты на разных языках, включая русский, и относящиеся к различным предметным областям.

OpenAI — Американская научно-исследовательская организация, занимающиеся разработками в области искусственного интеллекта.

Актуальность: Актуальность разработки десктопного приложения для генерации текста на основе ChatGPT обусловлена возрастающим интересом к искусственному интеллекту в различных сферах, включая образование, маркетинг и креативные индустрии. С увеличением объемов информации и потребностью в быстром создании качественного контента, такие инструменты становятся необходимыми. Десктопные приложения предлагают удобный интерфейс, возможность работы с большими объемами данных и повышенный уровень контроля над личными данными, что делает их привлекательными для пользователей.

Объект исследования: Объектом исследования является процесс генерации текста с использованием технологий искусственного интеллекта, в частности, алгоритмов, основанных на модели ChatGPT.

Предмет исследования: Предметом исследования является функционал и пользовательский опыт десктопного приложения для генерации текста, а также его интеграция с другими инструментами и платформами.

Цель данной проектной работы: Целью данной проектной работы является разработка концепции и прототипа десктопного приложения, которое использует возможности ChatGPT для генерации текста, соответствующего потребностям пользователей в различных сферах.

Задачи:

1) Разработка функциональных требований: Определить ключевые функции и возможности приложения, включая интерфейс, настройки и интеграцию с другими инструментами.

2) Создание прототипа: Разработать прототип десктопного приложения, учитывая пользовательский интерфейс и опыт.

3) Документация: Подготовить полную документацию, включая описание функционала, инструкции по использованию и рекомендации по дальнейшему развитию приложения.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Терминология по проекту и глоссарий

**Основные термины:**

**1) GPT (Generative Pre-trained Transformer)** – модель машинного обучения от OpenAI, основанная на архитектуре Transformer. Использует предобучение на больших текстовых данных и fine-tuning для генерации осмысленного текста.

**2) API (Application Programming Interface)** – интерфейс для взаимодействия между программами. В GPT API позволяет подключать модель к сторонним сервисам.

**3) UI (User Interface)** – визуальная часть приложения, отвечающая за взаимодействие с пользователем. Хороший UI интуитивен и удобен.

**4) UX (User Experience)** – общее впечатление пользователя от работы с приложением, включая удобство, скорость и функциональность.

**Технические аспекты:**

**1) Модуль** – независимый компонент системы, выполняющий конкретную задачу (например, генерация текста).

**2) Алгоритм** – последовательность шагов для решения задачи (например, обработка запроса в GPT).

**3) Тренировка модели** – обучение нейросети на данных для улучшения её предсказаний. Включает pre-training (общее обучение) и fine-tuning (специализация).

**4) Контекст** – входные данные, на основе которых модель генерирует ответ. Чем он точнее, тем релевантнее результат.

**5) Параметры генерации** – настройки вывода (длина, температура – мера "креативности").

**Дополнительные функции:**

**1) Логирование** – запись событий для отладки и анализа работы системы.

**2) Сохранение состояния** – возможность продолжить работу с предыдущими данными (например, история чата).

**3) Обратная связь (Feedback)** – оценки и комментарии пользователей для улучшения продукта.

**4) Кроссплатформенность** – поддержка разных ОС (Windows, macOS, Linux) за счёт универсального кода.

**5) Безопасность данных** – защита информации (шифрование, аутентификация).

**6) Документация** – руководства для пользователей и разработчиков.

**Как работает GPT?**

GPT использует **трансформерную архитектуру** с механизмом внимания (attention), что позволяет анализировать контекст и генерировать связный текст. Модель предобучается на огромных корпусах текстов, а затем дорабатывается под конкретные задачи (например, чат-бот). Чем больше параметров (например, GPT-3 имеет 175 млрд), тем лучше качество генерации, но выше требования к вычислительным ресурсам

## 1.2 Распределение ролей и работы

По части разработки UI UX интерфейс вид md работали Семенов Юрий и Слепцов Евгений:

**1. Разработка дизайна интерфейса:**

* Создали визуальную концепцию в **Figma** в стиле Material Design 3

(MD3).

* Подобрали цветовые палитры (основные, фоновые и акцентные

цвета).

* Определили типографику (шрифты, размеры текста, межстрочные расстояния).
* Продумали анимации: плавные переходы между экранами, микровзаимодействия с кнопками.

2. Разработка элементов интерфейса:

* Спроектировали компоненты:

**1) Карточки сообщений** — для отображения диалогов  
**2) Поля ввода** — с удобной клавиатурой и подсказками.  
**3) Кнопки** — плавающая кнопка отправки, кнопки настроек.  
**4) Навигация** — нижнее меню для быстрого перехода между разделами.

3. Раздел «О себе»:

* Добавили страницу с:

Описание проекта – цели, технологии

Информацией о разработках – фото, контакты

Ссылками на GitHub и соцсети

По части разработки модели Chatgpt c g4f работали Цвикальский Артём и Лаптев Никита:

1. Интеграция GPT-модели:

* + Настроили работу с бесплатным API g4f (аналог GPT-3.5/4)
  + Реализовали отправку запросов к модели и получение ответов
  + Добавили обработку ошибок(например, при отсутствии интернета)

2. Интерфейс чата:

* + Разработали экран переписки с:
* Историей сообщений
* Поле ввода
* Кнопка отправки

1. Настройки приложения:

* Выбор языка – русский/английский(с мгновенным применением)

По части создании Базы Данных работал Степанов Руслан:

1. Создание базы данных:

* + Разработал структуру БД для хранения:

Истории переписки (вопросы пользователя и ответы GPT)

Настроек приложения (тема, язык и другие параметры)

2. Подключение к программе

* + Настроили автоматическое сохранение всех сообщений в БД
  + Реализовали быструю загрузку последних диалогов при запуске

По части разработки настроек и о себе для интерфейса работали Захаров

3. Дополнительно:

* Реализовали тёмную тему с учётом рекомендаций Material Design
* Настроили локализацию(все тексты подгружаются в зависимости от выбранного языка).

## 1.3 Стек Технологий

Мы будем использовать такие Стек-технологии в проекте как:

1. Python

Основной язык разработки. Простота, богатые библиотеки, кроссплатформенность

1. Tkinter

Графический интерфейс(GUI). Плюсы: Встроен в Python, легковесный, подходит для простых приложений

1. Request

HTTP-запросы к API(например, для ChatGPT). Пример: Отправка запросов и обработка ответов от сервера

1. SQL Server

Хранение данных(история чатов, настройки). Альтернатив: SQLite(Если нужна локальная БД без сервера)

1. GitHub

Хранение кода

Совместная работа (ветки, pull requests)

Отслеживания задач (issues. Projects)

1. VS Code

Лёгкий и мощный редактор

Поддержка Python (плагины, отладка)

Интеграция с GitHub

1. ChatGPT

Генерация ответов на сообщения

Возможность интеграции через OpenAI API или же(что мы и используем) бесплатные аналоги – g4f

# 2. ПРАКТИЧЕСКАЯ ЧАСТЬ (РАЗРАБОТКА)

## 2.1 Архитектура ПО

Десктопное приложение для генерации текста на основе ChatGPT построено по модульной архитектуре, включающей следующие компоненты:

Графический интерфейс (на базе Tkinter): обеспечивает взаимодействие пользователя с приложением через три вкладки: "Чат" (для переписки), "Настройки" (для выбора языка) и "О нас" (информация о разработчиках).

Модуль интеграции с ChatGPT (на базе библиотеки g4f): отвечает за отправку пользовательских запросов к модели gpt-4o-mini и получение ответов.

Модуль базы данных (на базе pyodbc и MSSQL): сохраняет историю сообщений (вопросы пользователя и ответы бота) и загружает её при запуске.

Модуль настроек: позволяет переключать язык интерфейса (русский/английский) с динамическим обновлением текстов.

Архитектура реализована с использованием объектно-ориентированного подхода в Python. Класс ChatApp объединяет все модули, обеспечивая их взаимодействие через методы для обработки сообщений, управления интерфейсом и работы с базой данных.

## 2.2 Разработка Проекта по ролям

Каждый участник команды внёс вклад в определённые аспекты проекта, что отражено в коде и описано ниже.

Семёнов Юрий и Слепцов Евгений: Разработка UI/UX

Семёнов Юрий и Слепцов Евгений отвечали за проектирование и реализацию пользовательского интерфейса в соответствии с принципами Material Design 3. Их задачи включали:

Создание вкладочной структуры с использованием ttk.Notebook для разделов "Чат", "Настройки" и "О нас".

Реализация текстового поля для отображения истории сообщений (scrolledtext.ScrolledText) и поля ввода (ttk.Entry) с кнопкой отправки.

Настройка локализации интерфейса с динамическим переключением языка (русский/английский) через метод update\_ui\_language.

Разработка вкладки "О нас" с информацией о команде.

Пример кода (настройка вкладки "Чат", метод setup\_chat\_tab) (рис 1)



Рисунок 1 Интерфейс

Этот код создаёт вкладку "Чат" с текстовым полем для истории сообщений, полем ввода и кнопкой отправки, обеспечивая удобное взаимодействие пользователя с чат-ботом.

Цвикальский Артём и Лаптев Никита: Интеграция с ChatGPT

Цвикальский Артём и Лаптев Никита реализовали модуль взаимодействия с моделью ChatGPT через библиотеку g4f. Их вклад:

Настройка отправки запросов к модели gpt-4o-mini и получение ответов (метод get\_ai\_response).

Реализация асинхронной обработки запросов с использованием threading.Thread для предотвращения зависания интерфейса.

Обработка ошибок API, таких как отсутствие интернета или сбои сервера.

Пример кода (интеграция с ChatGPT, метод get\_ai\_response) (рис.2)



Рисунок 2 Модель GPT Chat’а

Этот код отправляет пользовательское сообщение к модели ChatGPT, получает ответ, сохраняет его в базе данных и обновляет интерфейс. Асинхронная обработка через threading.Thread и self.root.after обеспечивает плавную работу приложения.

Степанов Руслан: Работа с базой данных

Степанов Руслан разработал модуль взаимодействия с базой данных MSSQL. Его задачи:

Создание структуры таблицы gptchat\_saves для хранения истории переписки (сообщения пользователя и ответы бота).

Реализация автоматического подключения к базе данных с перебором ODBC-драйверов (метод create\_connection).

Сохранение сообщений (save\_message) и загрузка истории при запуске (load\_history).

Пример кода (подключение к базе данных, метод create\_connection) (рис.3)



Рисунок 3. База данных MS SQL

Этот код обеспечивает надёжное подключение к MSSQL, перебирая доступные драйверы и обрабатывая ошибки подключения.

## 2.3 Контроль выполнения плана

Разработка проекта велась с использованием GitHub для управления версиями кода и задачами. Этапы работы:

Неделя 1: Определение функциональных требований и прототипирование интерфейса в Figma.

Недели 2–3: Разработка модулей: интерфейс (Семёнов, Слепцов), интеграция с ChatGPT (Цвикальский, Лаптев), база данных (Степанов).

Неделя 4: Тестирование приложения, включая проверку работы чата, сохранения сообщений и переключения языка.

Неделя 5: Подготовка документации и финальных правок.

Команда проводила регулярные встречи для синхронизации прогресса, используя GitHub Issues для отслеживания задач и Pull Requests для проверки кода.

# ЗАКЛЮЧЕНИЕ

В результате работы команда ОГУЗКИ разработала десктопное приложение для генерации текста на основе модели ChatGPT с использованием библиотеки g4f. Приложение, реализованное на Python, включает интуитивный интерфейс на базе Tkinter с вкладками "Чат", "Настройки" и "О нас", интеграцию с моделью gpt-4o-mini для генерации ответов, а также модуль взаимодействия с базой данных MSSQL для хранения истории переписки. Реализована поддержка переключения языка интерфейса (русский/английский). Приложение обеспечивает стабильную работу, сопоставимую с веб-версией ChatGPT, и позволяет сохранять диалоги для дальнейшего использования.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Microsoft. Справочник по Python. URL: https://docs.microsoft.com/en-us/python

Документация библиотеки g4f. URL: https://github.com/xtekky/gpt4free

Документация Tkinter. URL: https://docs.python.org/3/library/tkinter.html

Документация pyodbc. URL: https://github.com/mkleehammer/pyodbc

# ПРИЛОЖЕНИЕ 1