**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**Отчёт по проектной (учебной) практике**

**Дисциплина:**  Проектная (учебная) практика

**Выполнил: студент группы**  241-336

Гужева Арина Игоревна

(Фамилия И.О.)

**Дата, подпись** 20.05.2025

(Дата) (Подпись

**Проверил:**

(Фамилия И.О., степень, звание) (Оценка)

**Дата, подпись**

(Дата) (Подпись)

**Замечания:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва**

**2025**

Оглавление

[Вариативная часть задания (Практическая технология производства). Создание Telegram бота с помощью языка программирования Python 3](#_Toc198942797)

[Теоретическая часть. Техническое задание на разработку Telegram-бота. 3](#_Toc198942798)

[Описание: 3](#_Toc198942799)

[Общие требования: 3](#_Toc198942800)

[Данные: 4](#_Toc198942801)

[Практическая часть. Этапы разработки 6](#_Toc198942802)

[Настройка окружения 6](#_Toc198942803)

[Создание Telegram бота 6](#_Toc198942804)

[Написание программы для бота 8](#_Toc198942805)

[3. Добавление конфигурации для запуска бота 14](#_Toc198942806)

[Дополнительная часть. Модификация проекта. 15](#_Toc198942807)

[Загрузка репозитория на гитхаб 17](#_Toc198942808)

[Заключение 17](#_Toc198942809)

[Список литературы: 18](#_Toc198942810)

# Вариативная часть задания (Практическая технология производства). Создание Telegram бота с помощью языка программирования Python

Теоретическая часть. Техническое задание на разработку Telegram-бота.

### Описание:

Телеграм бот “FilmBot” предназначен для подбора 1-3 фильмов по запросам пользователя. Бот предлагает варианты ответа по трём критерием (жанр, настроение, длительность) и на основании ответов определяет, какой пункт подходит пользователю и выводит его на экран.

### Общие требования:

1. Стековые технологии:
2. Язык программирования: Python
3. Среда программирования: PyCharm
4. Хостинг: GitHub
5. Создание Markdown файла: VS Code
6. Расширение: Markdown All in One

Функциональные требования:

|  |  |  |
| --- | --- | --- |
| Команда | Описание | Содержание |
| /start | Начинает диалог с пользователем | Привет, (имя пользователя) Я подберу фильм по твоим предпочтениям ❤️\n\n'  "Выбери один из основных жанров в меню: |

### Данные:

1. База данных с информацией про фильмы.

movies\_db = [

{"title": "Крестный отец", "genre": "криминал", "mood": "напряжённый", "duration": "длинный"},

{"title": "Побег из Шоушенка", "genre": "драма", "mood": "добрый", "duration": "длинный"},

{"title": "Темный рыцарь", "genre": "боевик", "mood": "экшн", "duration": "длинный"},

{"title": "Форрест Гамп", "genre": "драма", "mood": "добрый", "duration": "длинный"},

{"title": "Начало", "genre": "фантастика", "mood": "психологический", "duration": "длинный"},

{"title": "Матрица", "genre": "фантастика", "mood": "экшн", "duration": "средний"},

{"title": "Джанго освобожденный", "genre": "боевик", "mood": "экшн", "duration": "длинный"},

{"title": "Бойцовский клуб", "genre": "драма", "mood": "психологический", "duration": "длинный"},

{"title": "Зеленая книга", "genre": "драма", "mood": "добрый", "duration": "средний"},

{"title": "Достать ножи", "genre": "триллер", "mood": "психологический", "duration": "средний"},

{"title": "Однажды в Голливуде", "genre": "драма", "mood": "романтический", "duration": "длинный"},

{"title": "Джокер", "genre": "драма", "mood": "психологический", "duration": "средний"},

{"title": "Паразиты", "genre": "триллер", "mood": "напряжённый", "duration": "средний"},

{"title": "Гравитация", "genre": "фантастика", "mood": "напряжённый", "duration": "короткий"},

{"title": "Мальчишник в Вегасе", "genre": "комедия", "mood": "весёлый", "duration": "средний"},

{"title": "Интерстеллар", "genre": "фантастика", "mood": "психологический", "duration": "длинный"},

{"title": "Волк с Уолл-стрит", "genre": "комедия", "mood": "весёлый", "duration": "длинный"},

{"title": "Остров проклятых", "genre": "триллер", "mood": "напряжённый", "duration": "длинный"},

{"title": "Гладиатор", "genre": "боевик", "mood": "экшн", "duration": "длинный"},

{"title": "Титаник", "genre": "драма", "mood": "романтический", "duration": "длинный"},

{"title": "Криминальное чтиво", "genre": "криминал", "mood": "напряжённый", "duration": "длинный"},

{"title": "Одержимость", "genre": "драма", "mood": "психологический", "duration": "средний"},

{"title": "Дедпул", "genre": "комедия", "mood": "весёлый", "duration": "средний"},

{"title": "Безумный Макс", "genre": "боевик", "mood": "экшн", "duration": "средний"},

{"title": "Оно", "genre": "триллер", "mood": "напряжённый", "duration": "длинный"}

]

1. Варианты жанров: боевик, драма, комедия, фантастика, криминал, триллер.
2. Варианты настроения: романтический, экшн, добрый, напряжённый, весёлый, психологический.
3. Варианты длительности: короткий, средний, длинный, любой.

Практическая часть. Этапы разработки

### Настройка окружения

* Установка языка программирования Python 3.9+
* Установка среды программирования PyCharm
* Создание нового проекта, название Bot\_practic

Подключение библиотеки в терминале библиотеки для установления зависимости:

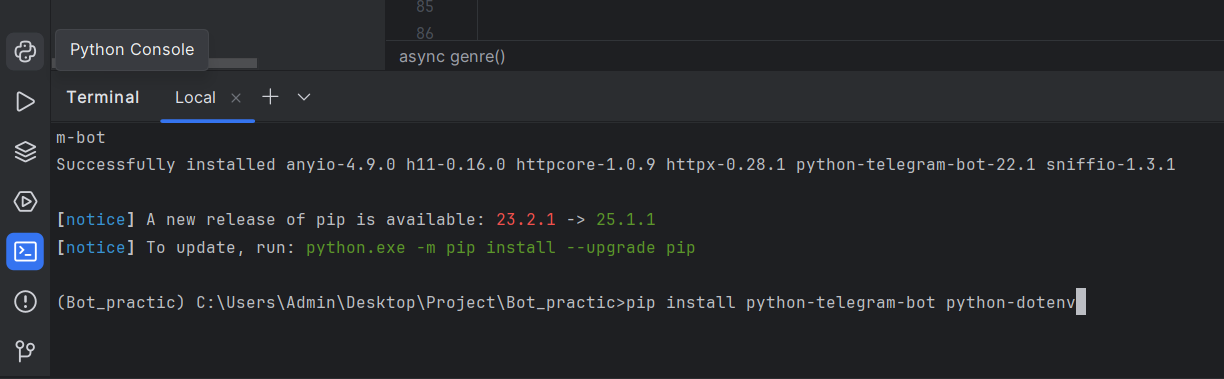


Рис. 1 - Установка библиотеки pip install python-telegram-bot

### Создание Telegram бота

Переходим непосредственно в приложение Telegram и находим @BotFather – официального бота для создания, оформления и редактирования ботов. Затем вводим программы /start для открытия меню и /newBot для создания своего бота. Далее придумываем название и юзернейм, после чего бот будет создан, мы получаем его токен (он будет использоваться в коде).

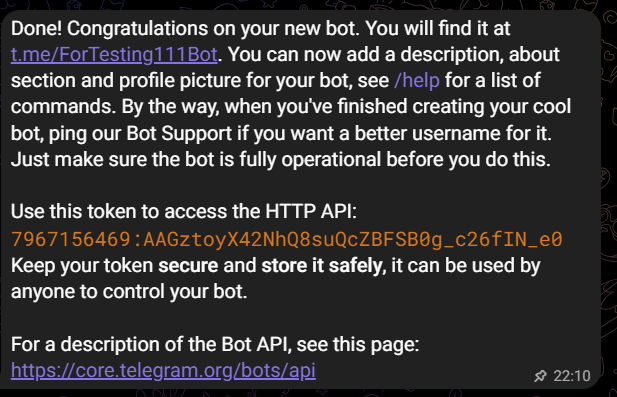
 

Рис. 2 - BotFather Рис.3. Токен созданного бота

Далее настраиваем фото и описание и получаем готовый внешний вид будущего бота.



Рис. 4 - Готовый внешний вид бота

### Написание программы для бота

Импортируем библиотеки:

* ReplyKeyboardMarkup - для удобного выбора вариантов
* ConversationHandler - для управления многоэтапным диалогом
* MessageHandler - для обработки текстовых сообщений

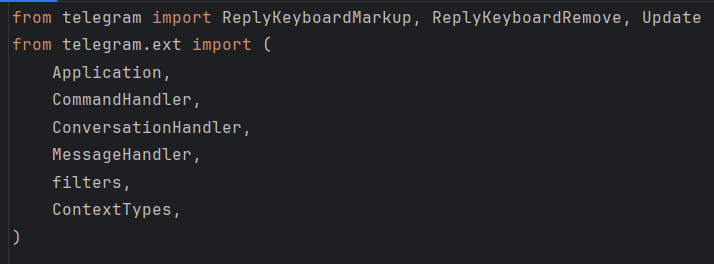


Рис. 5 - Импорт библиотек

Определяем состояния диалога:

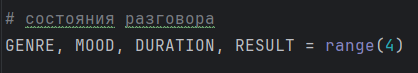


Рис. 6 - Состояние диалога

Создание базы данных фильмов на основании ТЗ:

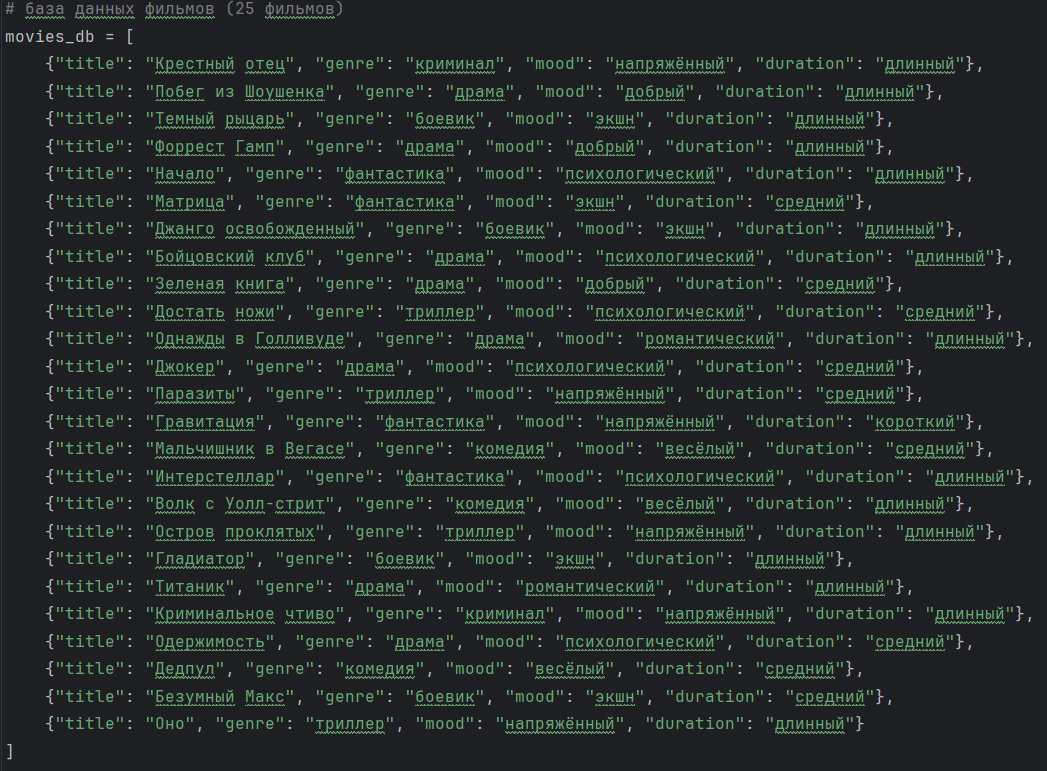


Рис. 7 - Сгенерированный список фильмов

Далее создаём функцию start(), в которой определяем имя пользователя:

user = update.effective\_user

Вводим меню вариантов жанров, причём оформляем в две строчки с помощью квадратных скобок для лучшего визуала:

reply\_keyboard = [["боевик", "драма", "комедия"],

["фантастика", "криминал", "триллер"]]

Далее приветствуем пользователя, выводя начальную строчку и переходим в состояние жанра:

return GENRE

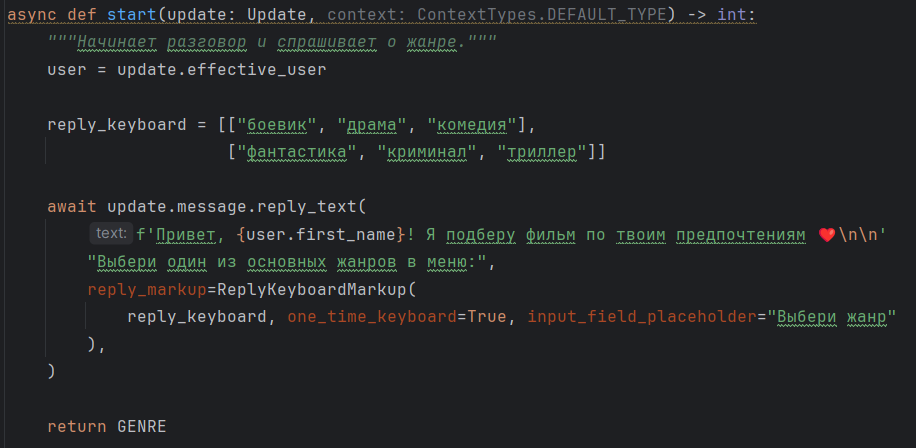


Рис. 8 - Функция start

Создаём функцию genre, в котором выводим меню с вариантами жанров, обрабатываем выбор и выводим строчку, обозначающую переход к следующему состоянию – настроению.

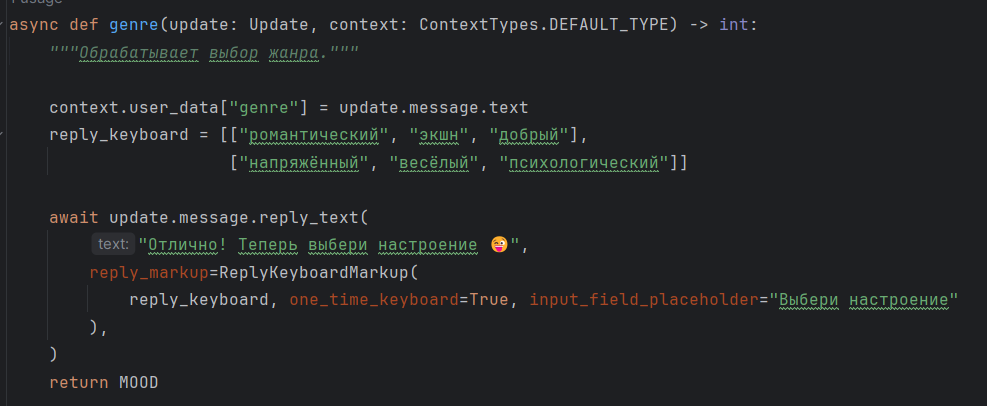


Рис. 9 - Функция genre

По аналогии создаём функцию mood() для определения настроения, меняя только данные.

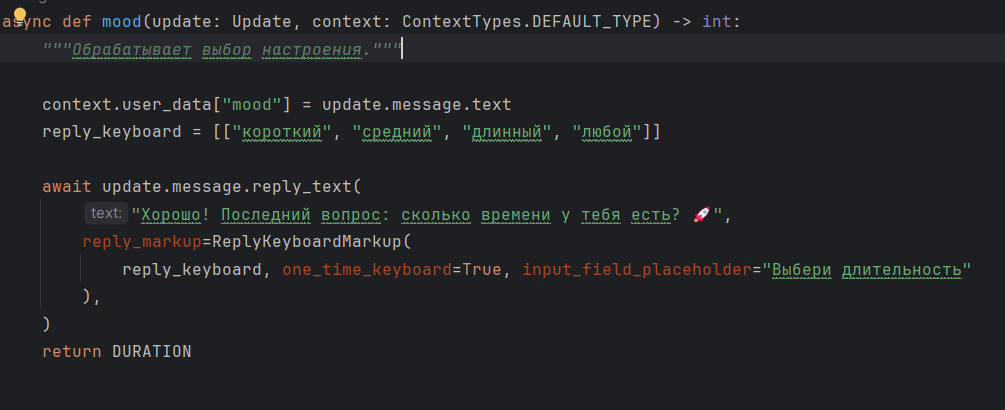


Рис. 10 - Функция mood

Функция Duration() повторяет элементы предыдущих функций:

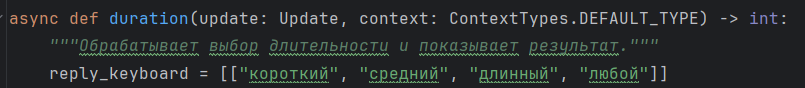


Рис. 11 - Функция duration

А также добавляются следующие элементы:

Присваивание значений выбранных вариантов:

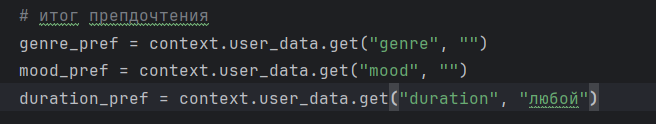


Рис. 12 - Функция duration

Цикл для подбора фильмов:

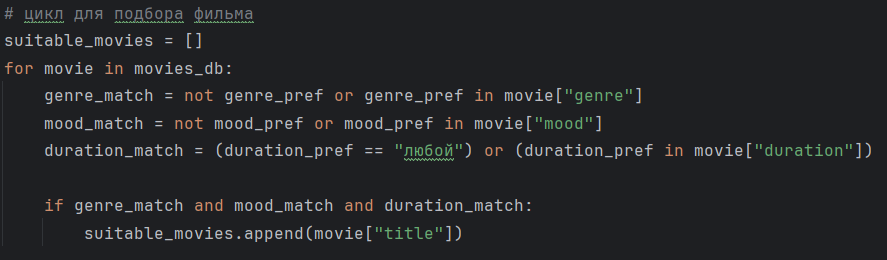


Рис. 13 – Цикл для подбора фильмов

Обработчик событий: в случае того, если фильм не будет найден – выводится случайный фильм:

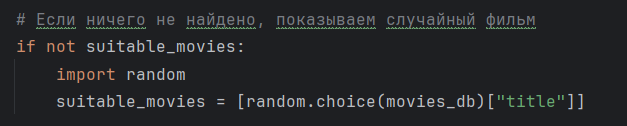


Рис. 14 – Обработчик событий

Ограничение вывода максимум 3-х фильмов:

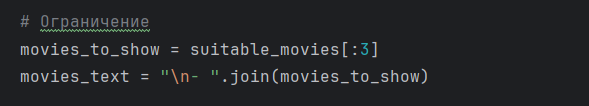


Рис. 15 – Ограничение

Вывод подобранного фильма:

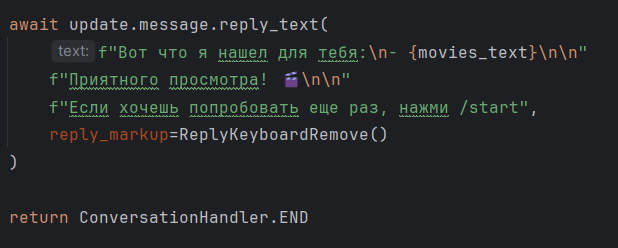


Рис. 16 – Обработка выбора

Далее создаём основную функцию main():

Вводим токен нашего бота и прикрепляем его к программе:

Application = Application.builder().token("7967156469:AAGztoyX42NhQ8suQcZBFSB0g\_c26fIN\_e0").build()

Вызываем функции в нужном порядке:



Рис. 17 – Функция main

Где компоненты:

Application — центральный класс библиотеки python-telegram-bot, управляющий всеми процессами.

builder() — конструктор для настройки бота.

token() — передача API-токена, полученного от @BotFather.

entry\_points — точки входа в диалог. Здесь — команда /start, которая вызывает функцию start().

states — словарь состояний диалога:

GENRE: Ожидание выбора жанра → вызывает genre().

MOOD: Ожидание выбора настроения → вызывает mood().

DURATION: Ожидание выбора длительности → вызывает duration().

/cancel — завершает диалог.

В конце добавляем небольшой цикл для работы программы:

if \_\_name\_\_ == "\_\_main\_\_":

main()

На этом написание программы заканчивается

### 3. Добавление конфигурации для запуска бота

Рядом с кнопкой запуска программы есть меню конфигураций, где необходимо создать новую конфигурацию

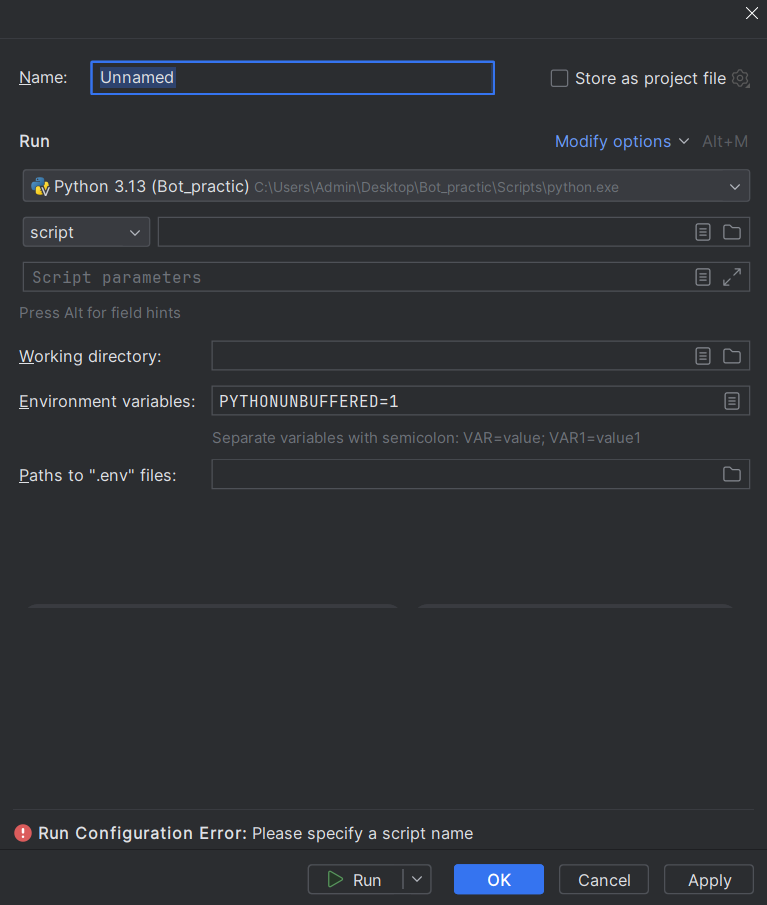


Рис. 18 – Создание новой конфигурации

Главное указать правильно путь к файлу проекта. После этого нажимаем на + и с помощью новой конфигурации запускаем бота.

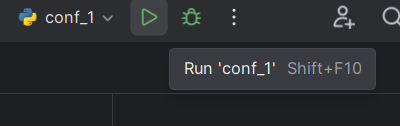


Рис. 19 – Запуск новой конфигурации

Результат работы:

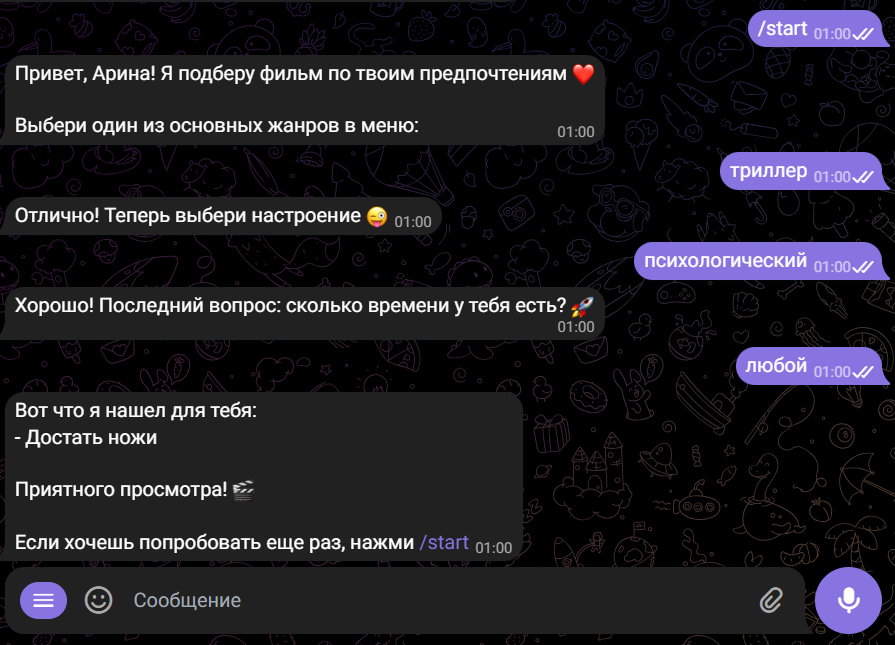


Рис. 20 – Запуск новой конфигурации

Дополнительная часть. Модификация проекта.

Для оптимизации работы бота добавим функцию обработки ошибок при введении текстовых сообщений пользователем, так как для корректной работы программы необходимо, чтобы пользователь только выбирал из предложенных вариантов.

Для этого создадим функцию для обработки текста, когда диалог не начат:

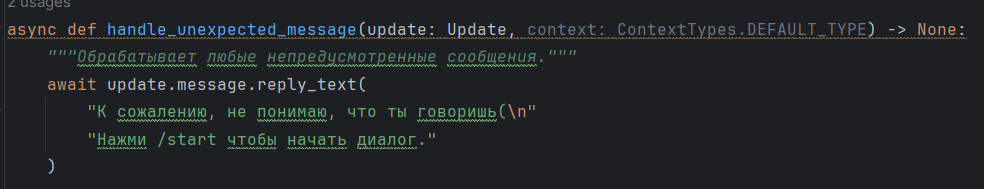


Рис. 21 – Функция обработки текста

Также добавим проверки в каждое состояние на проверку того, чтобы ответ пользователя были один из вариантов:

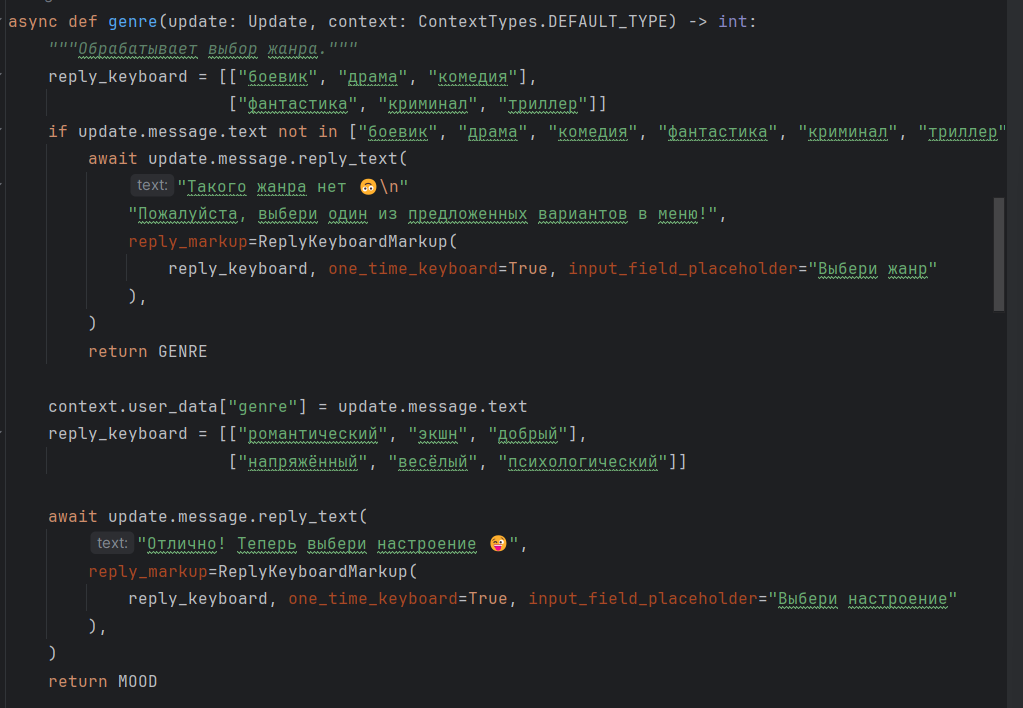


Рис. 22 – Модификация функций (добавление проверок)

Вот редактированная функция для жанра. По аналогии добавляем обработку в настроение и время. В итоге получаем модифицированного бота.

Результат:

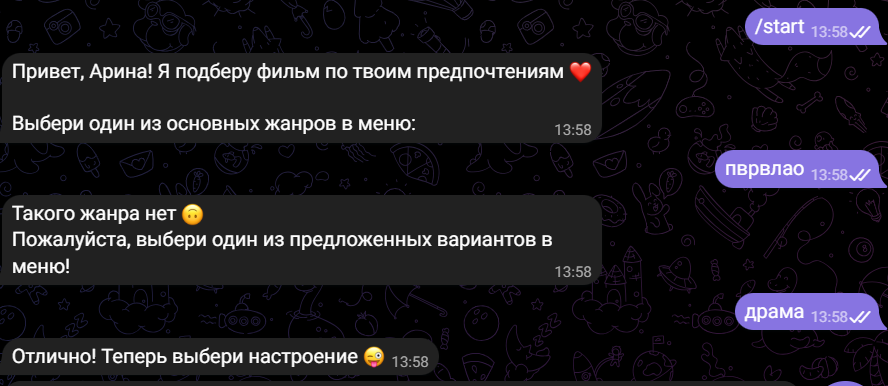


Рис. 23 – Результат модификации

Загрузка репозитория на гитхаб

Для этого необходимо зарегистрироваться на официальном сайте гитхаба и создать новый репозиторий:

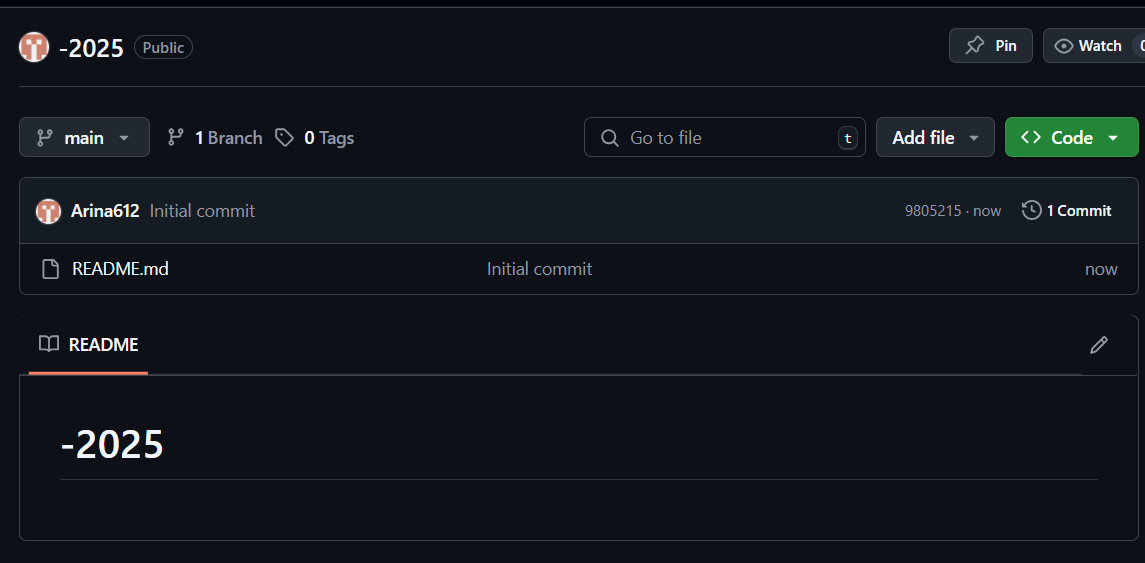
****

Рис. 24 – Размещение на гитхаб

Затем оформляем структуру и добавляем нужные файлы. Получаем готовый проект.

Заключение

Для выполнения вариативной части проектной практики 2025 были изучены и применены на практике материалы для создания телеграмм бота. Языком программирования служил Python, а также использовались библиотеки для работы с приложением Телеграмм. Работа направлена на усвоение нового материала в программировании, умению работать с библиотекой aiogram и другими, создавании и настроек бота в самом приложении, работой с гитхабом и оформлением отчёта в виде Markdown формата.

В итоге полученные навыки можно совершенствовать и применять в будущем не только в учебной, но и в профессиональной деятельности.

Ссылка на гитхаб:

<https://github.com/Arina612/->

Список литературы:

1. Как создать телеграм-бота на Python: инструкция [электронный ресурс]. Режим доступа: <https://timeweb.cloud/tutorials/python/kak-sozdat-telegram-bota-na-python>

(дата обращения: 10.05.25)

1. Как создать бота в Telegram: большая инструкция [электронный ресурс]. Режим доступа: <https://blog.skillfactory.ru/kak-sozdat-bota-v-telegram/>

(дата обращения: 10.05.25)

1. Чат-боты в Telegram на Python и Aiogram: пишем первого бота [электронный ресурс]. Режим доступа: <https://timeweb.com/ru/community/articles/chat-boty-v-telegram-na-python-i-aiogram-pishem-pervogo-bota>

(дата обращения: 10.05.25)

1. Как залить проект на GitHub бота [электронный ресурс]. Режим доступа: <https://skillbox.ru/media/code/instruktsiya_zalivaem_proekt_na_github_bez_komandnoy_stroki/#stk-2>

(дата обращения: 10.05.25)

1. За границей Hello World: полный гайд по разработке Telegram ботов с помощью Python и Aiogram 3. Часть 1 [электронный ресурс]. Режим доступа: <https://habr.com/ru/articles/732136/>

(дата обращения: 10.05.25)

1. ТОП-3 лучших киноботов для просмотра фильмов в Telegram [электронный ресурс]. Режим доступа: <https://pikabu.ru/story/top3_luchshikh_kinobotov_dlya_prosmotra_filmov_v_telegram_10041176>

(дата обращения: 10.05.25)