ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ МОЛДОВЫ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

**Михайлов Пётр** (студента)

**Индивидуальная работа**

по дисциплине „ПРОГРАММИРОВАНИЕ В PYTHON”

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия, имя, научная степень

(подпись)

Автор: \_\_\_\_Михайлов\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Кишинев, 2024

**Краткое описание языка Python:**

Python - это высокоуровневый язык программирования общего назначения, известный своей простотой и читаемостью. Он поддерживает несколько парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Python обладает богатой стандартной библиотекой и большим количеством сторонних модулей, что делает его подходящим для широкого спектра задач, от веб-разработки до анализа данных и машинного обучения.

**Краткая постановка задачи:**

Целью данной работы является создание телеграм-бота, который будет выполнять следующие функции:

1.Предоставление пользователям программ тренировок на каждый день недели.

2.Отправка пользователям случайных мотивационных цитат.

3.Предоставление списка музыкальных исполнителей для тренировок.

4.Обработка неизвестных команд и предоставление помощи пользователям.

**Логика реализованных алгоритмов:**

Целью данной работы является создание телеграм-бота, который будет выполнять следующие функции:

1.Предоставление пользователям программ тренировок на каждый день недели: Бот должен хранить и предоставлять пользователям различные программы тренировок в зависимости от выбранного дня недели. Пользователь выбирает день недели с помощью встроенной клавиатуры, и бот отправляет ему соответствующую программу тренировки.

2.Отправка пользователям случайных мотивационных цитат: Бот должен иметь возможность отправлять случайные мотивационные цитаты из заранее подготовленного списка. Пользователь может запросить цитату с помощью соответствующей команды.

3.Предоставление списка музыкальных исполнителей для тренировок: Бот должен предоставлять пользователям список музыкальных исполнителей, чья музыка может быть полезна и мотивирующая во время тренировок. Пользователь может запросить этот список с помощью соответствующей команды.

4.Обработка неизвестных команд и предоставление помощи пользователям: Бот должен обрабатывать команды от пользователей, предоставляя помощь и информацию о доступных командах. Если пользователь отправляет неизвестную команду, бот должен информировать его о доступных командах и их назначении.

5.Создание интерактивного интерфейса с помощью встроенной клавиатуры: Для удобства выбора дня недели бот должен использовать встроенную клавиатуру Telegram, содержащую кнопки с днями недели. Пользователь может выбрать день недели, нажав на соответствующую кнопку, и бот отправит программу тренировки для выбранного дня.

**Логика реализованных алгоритмов:**

Основные функции бота:

1.Инициализация бота

Библиотека telebot используется для создания объекта бота с помощью API ключа.

Импортируются необходимые модули: telebot для взаимодействия с Telegram API и random для генерации случайных чисел.

2.Хранение данных

Программы тренировок хранятся в виде словаря, где ключами являются дни недели, а значениями - списки упражнений для каждого дня.

Мотивационные цитаты хранятся в списке строк.

Список музыкальных исполнителей также хранится в виде списка строк.

3.Генерация меню с днями недели

Для создания интерактивного интерфейса используется InlineKeyboardMarkup.

Меню содержит кнопки с днями недели, каждая из которых при нажатии вызывает соответствующую функцию обработки.

4.Обработка команд и сообщений

Команда /start: Отправляет приветственное сообщение и выводит меню с днями недели. Сообщение содержит информацию о доступных командах, таких как /start, /help, /quote, и /music.

Команда /help: Отправляет сообщение с информацией о доступных командах и их назначении, помогая пользователям понять, как использовать бота.

Команда /quote: Отправляет случайную мотивационную цитату из заранее подготовленного списка, чтобы мотивировать пользователей.

Команда /music: Отправляет список музыкальных исполнителей, которые могут быть полезны для прослушивания во время тренировок.

Обработка нажатий на кнопки с днями недели: Когда пользователь нажимает на кнопку с названием дня недели, бот отправляет программу тренировки на этот день. Программа выбирается из словаря workouts.

Обработка неизвестных команд: Если пользователь отправляет команду, которую бот не распознает, бот отправляет сообщение с информацией о доступных командах и их назначении.

5.Дополнительные функции

Генерация случайных элементов: Для выбора случайной мотивационной цитаты используется функция random.choice, которая выбирает случайный элемент из списка motivational\_quotes.

Интерактивные элементы: Для создания кнопок меню используются InlineKeyboardButton, которые позволяют пользователям легко выбирать нужный день недели.

6.Обработка событий

Обработка команд: Команды обрабатываются с помощью декораторов @bot.message\_handler. Каждый декоратор связывает команду с соответствующей функцией обработки.

Обработка нажатий на кнопки: Нажатия на кнопки обрабатываются с помощью декоратора @bot.callback\_query\_handler. Функция обработки получает данные о нажатой кнопке и отправляет соответствующую программу тренировки.

7.Запуск бота

Бот запускается в режиме постоянного опроса сервера Telegram с помощью функции bot.polling(), что позволяет ему непрерывно обрабатывать команды и нажатия на кнопки.

Эта логика обеспечивает простую и интуитивно понятную работу бота, позволяя пользователям легко получать нужную информацию и программы тренировок.

**Созданные графические интерфейсы и их назначение:**

Для взаимодействия с пользователями телеграм-бот использует интерактивное меню с кнопками. Эти кнопки позволяют пользователям выбирать дни недели для получения соответствующих программ тренировок.

1.Меню с днями недели:

Назначение: Предоставить пользователю возможность выбора конкретного дня недели для получения программы тренировки.

Описание: В меню отображаются кнопки с названиями дней недели. Пользователь может нажать на любую из этих кнопок, чтобы получить программу тренировок для выбранного дня.

2.Сообщения с командами:

Назначение: Информировать пользователей о доступных командах и их назначении.

Описание: Когда пользователь вводит команду, бот отправляет сообщение с информацией о возможностях бота. Например, команды /start, /help, /quote, и /music предоставляют соответствующие сведения и инструкции.

3.Сообщения с ответами на команды:

Назначение: Предоставлять пользователю запрошенную информацию, такую как мотивационные цитаты или список музыкальных исполнителей.

Описание: Бот отправляет текстовые сообщения с мотивационными цитатами или списками музыкальных исполнителей в ответ на соответствующие команды пользователя.

Эти графические интерфейсы делают взаимодействие с ботом удобным и интуитивно понятным, позволяя пользователям легко получать нужную информацию и программы тренировок.

**Структуры данных с которыми я работал в приложении:**

Я использовал несколько структур данных в приложении. Для хранения программ тренировок на каждый день недели был использован словарь, где ключами являются названия дней недели, а значениями — списки упражнений. Мотивационные цитаты были сохранены в виде списка строк, каждая из которых представляет собой отдельную цитату. Для хранения списка музыкальных исполнителей, которых можно рекомендовать для прослушивания во время тренировок, я также использовал список строк, где каждая строка содержит имя исполнителя. Эти структуры данных позволили эффективно организовать и обрабатывать информацию, необходимую для работы бота.

**Использованные библиотеки:**

В проекте я использовал следующие дополнительные модули и библиотеки:

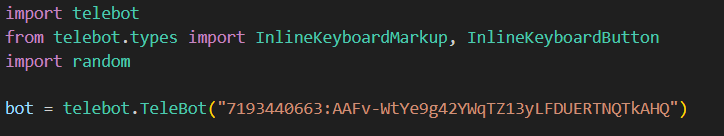
1) telebot: Это основная библиотека, используемая для взаимодействия с Telegram API. Она предоставляет удобные инструменты для создания ботов, обработки сообщений и нажатий кнопок. С помощью telebot я реализовал функции для отправки сообщений, создания интерактивного меню и обработки пользовательских команд.

2) random: Это стандартная библиотека Python для генерации случайных чисел и выбора случайных элементов из последовательностей. Я использовал модуль random для случайного выбора мотивационных цитат из списка, чтобы разнообразить ответы бота и поддерживать интерес пользователей.

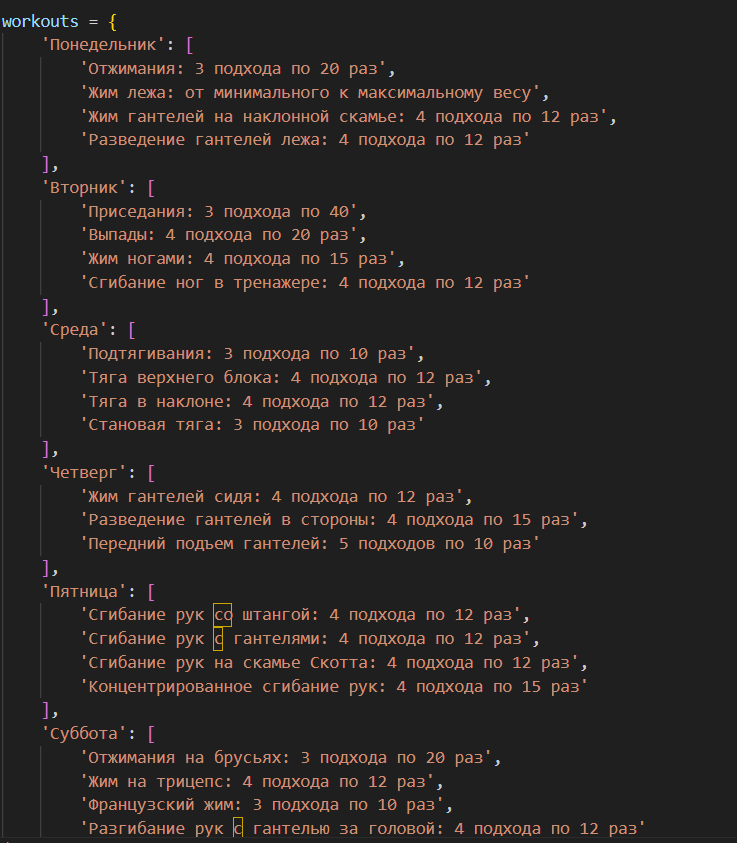
**Описание функционала приложения:**

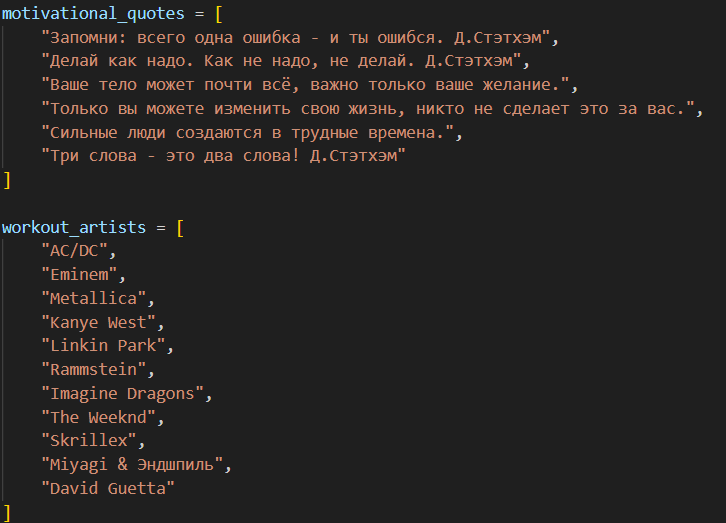
Данное мини-приложение, реализованное в виде телеграм-бота, служит личным тренером для пользователей, предоставляя программы тренировок, мотивационные цитаты и рекомендации по музыкальным исполнителям для тренировок. Пользователь взаимодействует с ботом через команды и интерактивное меню, что делает процесс получения информации удобным и интуитивно понятным. Основные цели использования бота: получение программ тренировок, получение мотивационных цитат и музыкальные рекомендации.

**Примеры кода:**

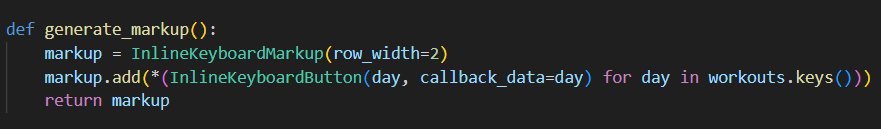
1. Здесь я импортировал необходимые библиотеки и создал объект TeleBot с использованием API ключа. Это позволяет боту взаимодействовать с серверами Telegram и обрабатывать сообщения от пользователей..

2. В этом блоке кода я создал структуры данных для хранения программ тренировок на каждый день недели, мотивационных цитат и имен музыкальных исполнителей. Это позволяет боту быстро и эффективно предоставлять пользователям необходимую информацию.

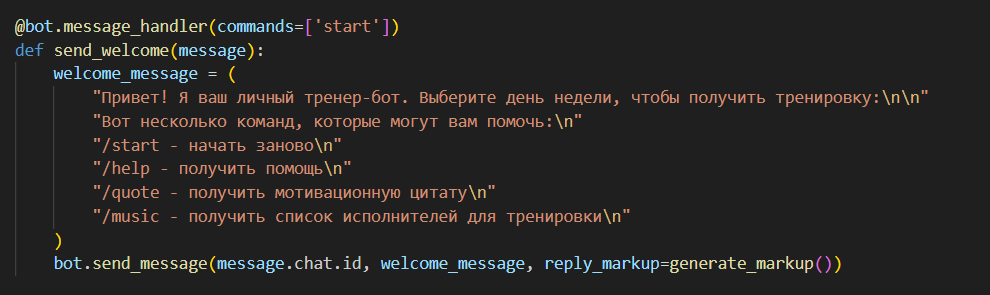




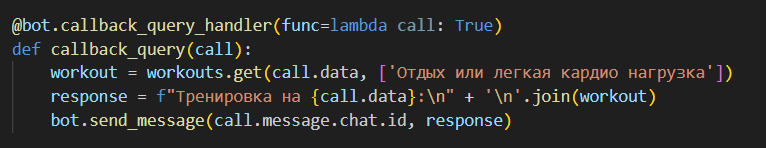
3. Здесь я создал функцию для генерации интерактивного меню с кнопками для каждого дня недели. Это меню позволяет пользователям выбирать день и получать соответствующую программу тренировок..



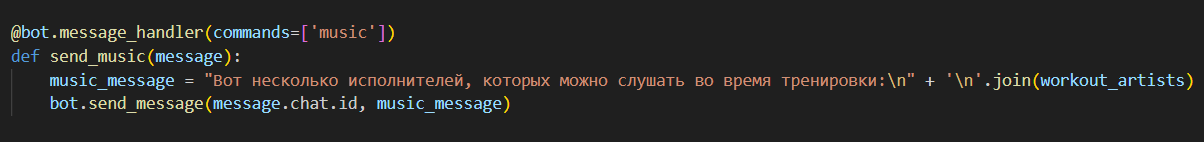
4. В этом коде я настроил команду /start, которая отправляет приветственное сообщение и отображает интерактивное меню. Пользователи видят приветствие и список доступных команд, что позволяет им легко начать взаимодействие с ботом.



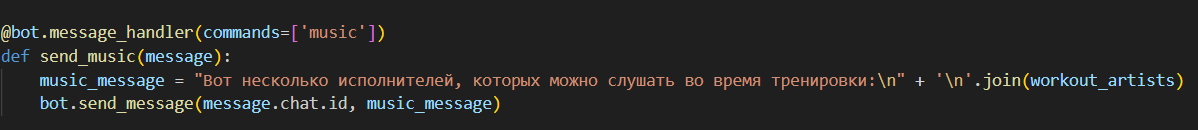
5. Здесь я настроил обработку нажатий на кнопки с днями недели, чтобы бот отправлял программу тренировки на выбранный день. Когда пользователь нажимает на кнопку с названием дня недели, бот извлекает соответствующую программу из словаря workouts и отправляет ее пользователю.



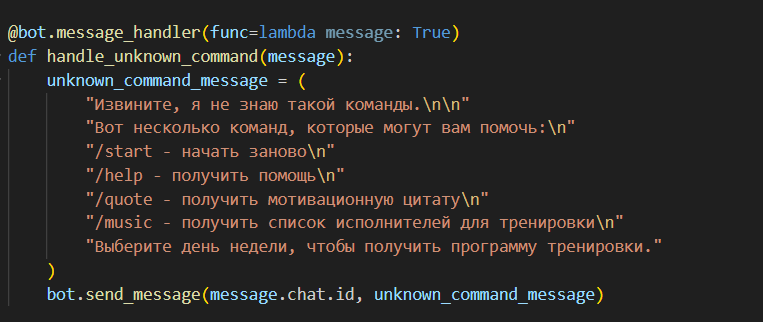
6. В этом кусочке кода я настроил команду /quote, которая отправляет случайную мотивационную цитату из списка. Это помогает пользователям получать вдохновение и поддерживать высокий уровень мотивации.



7. Здесь я настроил команду /music, которая отправляет список рекомендованных музыкальных исполнителей. Это помогает пользователям находить подходящую музыку для прослушивания во время тренировок.

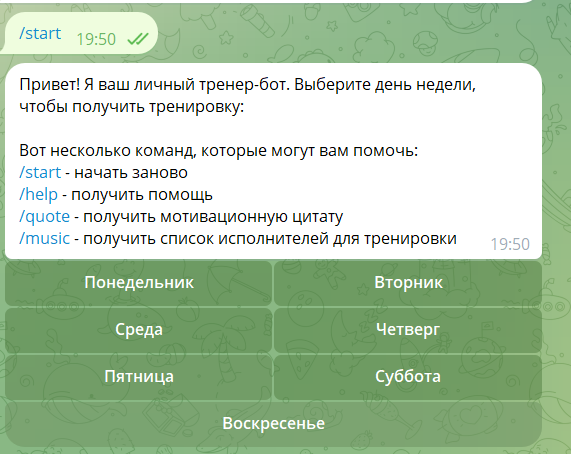


8. В данном кусочке кода я создал вывод ошибки на случай, если пользователь введет неверную команду, которая не существует

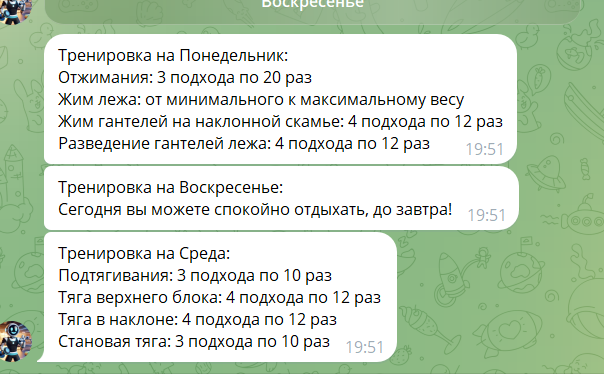


**Print-Screen:**

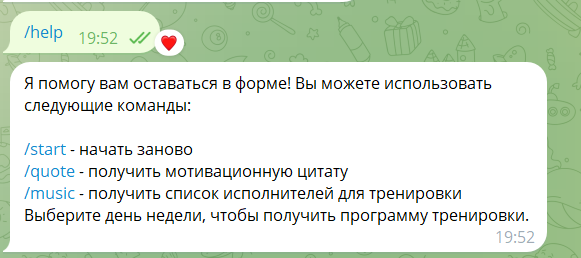
1.Стартовое меню при вводе команды /start:



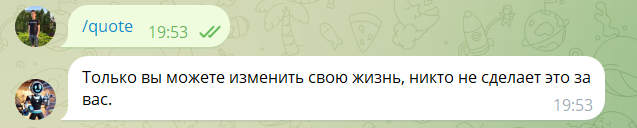
2.Нажав на нужный мне день, выводится упражнения, которые необходимо сделать в данный день.



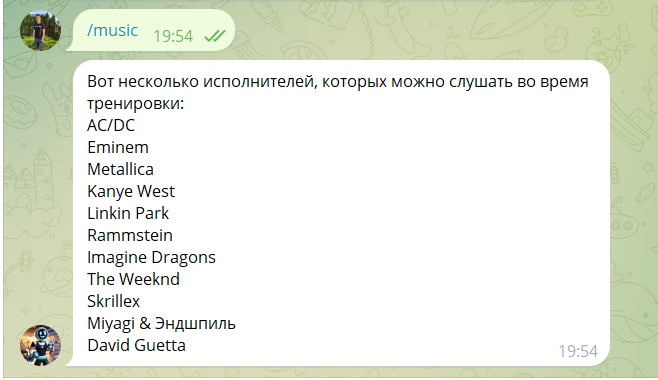
3.При вводе команды /help бот отвечает данным сообщением и показывает все возможные команды, которые ему можно ввести.



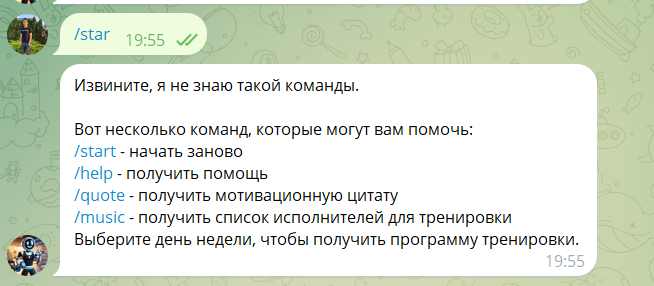
4.Вывод мотивационных цитат происходит благодаря команде /quote и выводится одна из цитат.



5. Написав команду /music бот нам выводит список исполнителей, которых можно послушать во время тренировки.



6. При вводе неизвестной команды, бот выдает ошибку и предоставляет команды, которые доступны.



**Вывод – как показался язык Python:**

Работа с Python в рамках этого проекта была приятным и продуктивным опытом. Язык Python показал себя как мощный и удобный инструмент для создания телеграм-бота. Благодаря его простоте и читаемости, я смог быстро написать и отладить код, а использование библиотеки telebot значительно упростило интеграцию с Telegram API. Создание словарей для хранения данных о тренировках и списков для мотивационных цитат и музыкальных исполнителей было интуитивно понятным и легко реализуемым. Язык Python позволил сосредоточиться на логике и функционале бота, не отвлекаясь на сложные синтаксические конструкции. В целом, Python отлично справился.