

Введение

Вы почти окончили курс и дошли до его завершающей части. Позади много пройденной теории и практики. Мы гордимся вашими трудолюбием и упорством! Осталось совсем немного — выполнить **финальную работу**.

Кто-то считает её формальностью, но давайте разберёмся, почему она нужна.

Во-первых, она станет первым кейсом для вашего портфолио, если вы только начинаете свой путь в профессии. Если продолжаете — выгодно его дополнит.

Во-вторых, работа наглядно покажет качество вашего кода. Глядя на неё, опытный разработчик оценит ваши навыки построения архитектуры приложения, стиль написания кода, в том числе нейминг, оформление, читаемость, гибкость. Это может стать поводом для первого собеседования. Всё в ваших руках.

В-третьих, телеграм-боты сейчас востребованы и широко используются для автоматизации рутинных процессов. Работая с ними, вы сможете заложить базу для развития будущих коммерческих проектов. Возможно, вы оставите своего первого бота как pet-проект или начнёте создавать ботов для себя и друзей, попытаетесь найти заказчика на биржах фриланса. Кто знает?

Поэтому давайте приступим. Будет интересно!

Тема финальной работы

Telegram-бот для работы с API стороннего сайта

Мы собрали основную информацию, чтобы вам было легче приступить к выполнению финальной работы.

Изучайте этот материал с карандашом в руках. Делайте пометки, выписывайте возникающие во время чтения вопросы, рисуйте блок-схемы, если необходимо. Словом, адаптируйте под свои потребности.

Постановка бизнес-задачи

К вам обратился заказчик, который планирует развивать бизнес в интернете. Для этого ему нужен первоклассный телеграм-бот. Однако его бюджет ограничен и не позволяет удовлетворить все его запросы.

После обсуждения деталей вы предложили универсальное решение — написать бота, у которого можно менять модуль работы с внешним API и при этом не трогать остальные модули.

Описание задачи

Вы можете выбрать любую тематику бота. Главное, чтобы:

- она вас привлекала;
- предусматривалась интеграция с API;
- сохранялась история запросов.

У вас есть возможность создать бота на свою тему или выбрать один из предложенных ниже.

У некоторых сайтов есть открытый, бесплатный API, над которым уже работали другие программисты, что позволяет вам запустить свои разработки на их основе.

Техническое задание на проект

Проект должен быть разработан на языке Python и быть портируемым. Это означает, что его можно запустить с помощью клонирования репозитория и установки необходимых библиотек следующей командой:

```
pip install -r requirements.txt
```

Подразумевается, что у заказчика уже установлен Telegram.

Для разработки проекта может быть использован любой открытый API. Обратите внимание, что некоторые API имеют ограничения на количество запросов, поэтому при выборе тематики бота обязательно изучите нюансы используемого API.

Например, на сайте rapidapi.com у базового пакета ограничение на количество запросов в месяц составляет от 50 до 500. Проблема нехватки запросов в данном случае решается дополнительной регистрацией одного или нескольких аккаунтов.

Описание внешнего вида и пользовательского интерфейса (UI)

При запуске Python-скрипта окно телеграм-бота должно воспринимать следующие команды:

- `/help` — для получения справки о доступных командах бота;
- `/history` — для вывода истории запросов пользователей;
- не менее двух команд, реализующих поиск согласно вашей идее, например:
 `/low` — для вывода минимальных показателей с соответствующим изображением товара/услуги и так далее; `/high` — для вывода максимальных показателей с соответствующим изображением товара/услуги и так далее.

Сообщение с результатом команды должно быть содержательным.

Технические требования

- Скрипт для телеграм-бота должен быть написан с использованием библиотек для работы с различными API и запросами.
- Функционал бота должен поддерживать работу со множеством пользователей и хранение их поисковой истории в базе данных с использованием [ORM Peewee](#).
- Запуск бота должен выполняться командой **python main.py** из терминала — из папки с проектом. Реализация main.py и остальных файлов проекта остаётся на ваше усмотрение. Файлы не должны содержать ошибок, работа должна быть корректной.
- Команды бота должны работать в соответствии с их описанием в техническом задании. Выдача результата производится по соответствующей команде телеграм-бота.
- Интерфейс должен быть отзывчивым — при возникновении пользовательских ошибок (например, ввод несуществующих команд или данных неверного типа) должно выводиться соответствующее уведомление.
- К скрипту должен быть приложен файл **readme.md**, который содержит инструкцию (документацию) для работы со скриптом и пользователем.

Обязательно согласуйте с куратором тематику и функционал до написания кода.

Идеи для проектов

Рекомендуем использовать собственные идеи, чтобы сделать ваш проект уникальным. Однако, если у вас нет конкретных представлений относительно того, какой проект выбрать, вы можете воспользоваться одним из предложенных ниже.

Тема №1. Поиск отелей

Ваша задача — разработать телеграм-бота, который будет взаимодействовать [с API отелей](#) и выполнять следующие команды:

- **lowprice** — отображение топа самых доступных отелей в указанном городе;
- **guest_rating** — предоставление информации о самых популярных отелях в выбранном городе;
- **bestdeal** — предоставление списка отелей, находящихся ближе всего к центру города;
- **history** — возможность просмотра истории запросов и поиска отелей.

В сценарий поиска должны быть включены следующие вопросы:

- получение города поиска;

- уточнение локации с помощью клавиатур;
- получение дат заезда/выезда;
- получение диапазона цен.

В вывод каждого отеля должны быть включены:

- название,
- ссылка на бронирование,
- описание,
- цена,
- выбранные даты,
- фотографии,
- координаты.

В вывод истории запросов должны быть включены:

- дата поиска,
- ссылка на бронирование,
- описание,
- цена,
- фотографии,
- координаты.

Будет плюсом, если:

- получение дат заезда/выезда будет реализовано с помощью клавиатур. Например, с использованием [календаря](#);
- вывод информации о каждом отеле будет реализован через [пагинацию](#);
- при запросе истории бот будет уточнять, за какую дату показывать историю.

Тема №2. Поиск фильмов

Вам нужно разработать телеграм-бота, который будет взаимодействовать [с API фильмов](#) и выполнять определённые команды:

- **movie_search** — поиск фильма/сериала по названию;
- **movie_by_rating** — поиск фильмов/сериалов по рейтингу;
- **low_budget_movie** — поиск фильмов/сериалов с низким бюджетом;

- **high_budget_movie** — поиск фильмов/сериалов с высоким бюджетом;
- **history** — возможность просмотра истории запросов и поиска фильма/сериала.

В сценарий поиска должны быть включены следующие вопросы:

- название;
- жанр (комедия, ужасы, фантастика и так далее);
- количество выводимых вариантов.

В вывод каждого фильма/сериала должны быть включены:

- название,
- описание,
- рейтинг,
- год производства,
- жанр,
- возрастной рейтинг,
- постер.

В вывод истории запросов должны быть включены:

- дата поиска,
- название фильма/сериала,
- описание фильма/сериала,
- рейтинг,
- год производства,
- жанр,
- возрастной рейтинг,
- постер.

Будет плюсом, если:

- при запросе истории бот будет уточнять, за какую дату показывать историю;
- вывод информации о каждом фильме/сериале будет реализован через [пагинацию](#);
- в истории поиска будут доступны кнопки для отметки просмотренных и несмотренных фильмов и сериалов.

Другие темы

Вы также можете взять за основу своего проекта одну из следующих идей для телеграм-бота:

- [Поиск рецептов блюд](#). Бот может предоставлять пользователю доступ к базе данных с рецептами блюд. Пользователь может искать рецепты по различным критериям, таким как тип блюда, ингредиенты, время приготовления и так далее.
- [Поиск футболистов](#). Бот может предоставлять информацию о футбольных игроках, их статистике, клубах, за которые они играют, текущих и прошлых командах, достижениях и прочее. Пользователи могут искать футболистов по различным критериям, таким как имя, позиция, клуб и так далее.
- [Поиск товаров с AliExpress](#). Бот может предоставлять возможность поиска товаров на платформе AliExpress. Пользователи могут искать товары по категориям, ключевым словам, ценовому диапазону и прочим критериям. Бот может также предоставлять информацию о характеристиках товаров, ценах, отзывах пользователей и так далее.

Формат сдачи материалов и оценивание

Рекомендуем постепенно разрабатывать проект, получая обратную связь от куратора на каждом этапе и внося необходимые корректировки. Однако, если вы уверены в своих силах, можете сразу прислать на проверку готовый проект.

Для разработки проекта используйте отдельный Git-репозиторий в gitlab.skillbox.ru.

Этапы разработки:

1. Создание бота, который реагирует на команду `/hello-world` и на текст «Привет» (необходимо сообщить проверяющему куратору имя вашего бота для тестирования).
2. Реализация команды 1.
3. Реализация команды 2.
4. Реализация команды 3.
5. Сдача проекта.

По завершении каждого этапа создайте свой Merge Request (MR) на ветку `master` и отправьте его куратору для код-ревью. Куратор оставит замечания или одобрит этот MR. В случае одобрения запрос сольётся в ветку `master`.

Рекомендуем создать первый MR с общей структурой проекта, в которой каждая команда телеграм-бота вынесена в отдельный файл. Это позволит вам в дальнейшем создавать несколько параллельных MR под разные функционалы из технического

задания, не ожидая одобрения куратора, и приступить к выполнению следующей задачи.

Таким образом, на этапе «Сдача проекта» вы сообщаете куратору о завершении проекта, в котором уже реализован весь необходимый функционал программы в соответствии с техническими требованиями.

Что оценивается

- Следование техническому заданию. Элементы должны работать в соответствии с описанием.
- Безошибочный запуск скрипта и корректная работа телеграм-бота.
- Отсутствие ошибок в результатах каждой реализованной команды.
- Usability приложения, то есть отзывчивость интерфейса. При наличии ошибок при вводе данных пользователь должен получать соответствующее уведомление.
- Чистота Python-кода: соблюдение правил [PEP8](#), описанная документация для классов и функций, наличие аннотаций типов, совместимость с **Python 3.10**.
- Наличие файла **readme.md** с инструкцией по эксплуатации скрипта и бота для пользователя.

Проект возвращается на доработку, если:

- Отсутствует или не работает базовая функциональность, описанная в техническом задании.
- Есть грубые ошибки в организации или структуре проекта.
- Куратор не может запустить проект локально.

Дополнительные материалы

Подсказки

- [«telebot быстро и понятно. Телеграмм-бот»](#) — статья о библиотеке telebot.
- [«Пишем ботов для Telegram на языке Python»](#) — статья о создании ботов и написании для них программ.

API и документация к ним

- [Выбирайте API на свой вкус](#)
- [Документация API Telegram bots](#)

Популярные библиотеки для разработки ботов

- [aiogram](#)
- [python-telegram-bot](#)
- [pyTelegramBotAPI](#)

Рекомендации по выполнению

- Внимательно изучите документацию используемых API. Протестируйте API на простых данных и проверьте корректность результата вручную на сайте.
- Начните с создания самого простого бота, реагирующего на одну-две команды, и добавьте короткий текст (например, «Привет»).
- Чтобы было легче выполнять проект, разбейте работу над ним на этапы. Например, начните с реализации первой команды и получения результата в самом скрипте. Затем добавьте вывод этого результата по команде бота и так далее. Часть работы может считаться готовой, если вы готовы продемонстрировать её заказчику. В вашем случае роль заказчика исполняет куратор.
- Составьте план работы по датам. Рекомендуем работать над финальным проектом продуктивно по два-три часа несколько дней в неделю вместо выполнения всего объёма работы за один подход. Не забывайте выделять время для отдыха.
- Отмечайте прогресс по мере выполнения плана. Это полезно по нескольким причинам: во-первых, вы будете держать ритм, во-вторых — контролировать ситуацию. И самое главное: каждый выполненный этап — это ваша маленькая победа. Чем больше таких побед вы будете отмечать, тем больше удовольствия получите от выполненного проекта.
- Прежде чем отправлять каждую часть проекта куратору (а в реальной работе — тестировщику), уделите время на проверку себя: посмотрите ещё раз код и результат, который он даёт. Идеально, если между завершением работы и проверкой себя пройдёт некоторое время — оно позволит вам переключиться и посмотреть на работу свежим взглядом.
- Качество продукта — это не метрика, а личное отношение разработчика. Пригласите нескольких друзей или коллег протестировать проект вместе с вами. Обращайте внимание на их поведение. Попросите проговаривать мысли, действия и ощущения. Задавайте наводящие вопросы. Вы можете заранее подготовить чек-лист, отталкиваясь от функциональных требований. Всё это поможет вам понять, в чём и на каких этапах у пользователя могут возникнуть проблемы, и сделать действительно удобный интерфейс.

Будет плюсом, если вы:

- Примените единый стиль кода с помощью [Black](#). Это инструмент для автоматического форматирования кода на Python. Он предлагает единое и строгое форматирование кода без возможности настройки. Black автоматически приводит код в соответствие с определённым стилем форматирования, что позволяет избежать споров о стиле кода в команде разработчиков. Посмотрите инструкцию о том, [как настроить Black в PyCharm](#).
- Настройте логирование, используя лог-файл в формате JSON. Например, для этих целей вы можете обратиться [к loguru](#).
- Используйте [паттерн проектирования MVC](#) (Model-View-Controller) для более удобной организации кода и разделения его на логические компоненты.
- Добавьте интерактивный функционал для улучшения пользовательского опыта. Это означает, что бот будет полноценно функционировать через inline-клавиатуру, минимизируя количество сообщений в чат.