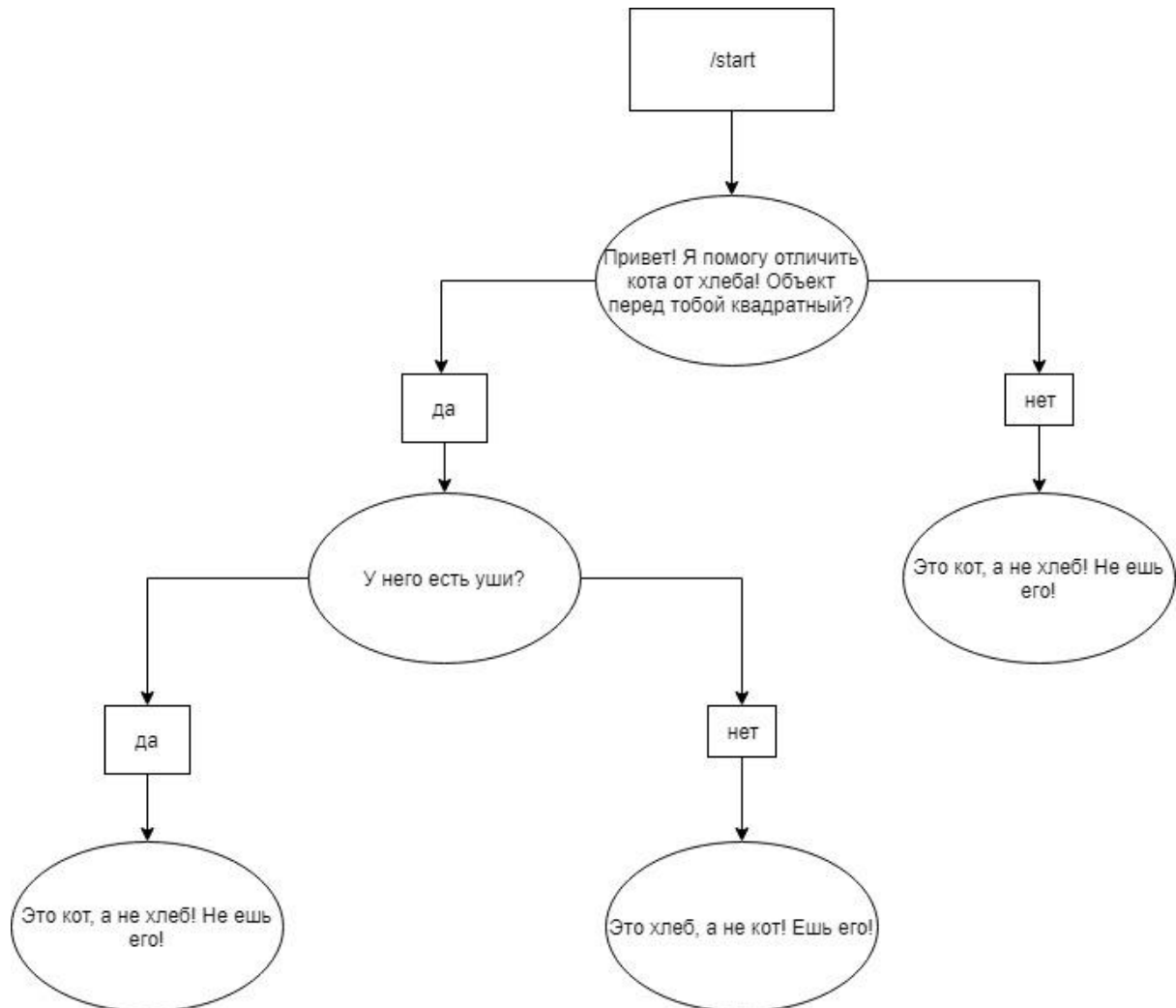


Задача 1.

Реализуйте бота, который поможет пользователю определить, кот перед ним или хлеб.
Сценарий бота следующий:



Сообщение `/start` должно всегда возвращать в начало диалога.

Бот должен интерпретировать как "да" сообщения, содержащие следующий текст:
"конечно", "ага", "пожалуй"

Бот должен интерпретировать как "нет" сообщения, содержащие следующий текст:
"нет, конечно", "ноуп", "найн"

Регистр сообщений от пользователя должен быть не важен.

Покройте бота тестами.

Спроектируйте и реализуйте HTTP API для общения с ботом: на вход надо принимать `id` пользователя (строку) и сообщение от пользователя (строку).

Заверните ваше решение в `docker`-образ.

Реализуйте сохранение истории переписки и сбора базовой статистики в БД

Напишите документацию: Описание API, инструкция по развёртыванию, схема БД.

Используйте Python 3.x с любыми библиотеками.

Что можно опционально сделать ещё, чтобы мы порадовались:

- 1) Поднять хостинг для бота и развернуть его на нём. Если CI поднимете, вообще будем в восторге.
- 2) Сделать распознавание ответов пользователя поумнее - исправлять опечатки или вообще какой-нибудь семантический поиск подрубить
- 3) Сделать распознавание кота или хлеба по фото (можно с использованием внешнего сервиса)

Формат решения: репозиторий на github/gitlab/bitbucket

Задача 2.

Хотим сделать бота, который отвечает на часто задаваемые вопросы. Для этого у нас есть какой-то чёрный ящик, который для данной реплики возвращает наиболее подходящий ответ и score, насколько по мнению этого чёрного ящика этот ответ подходит (от 0 до 100). При этом мы понимаем, что бот не сможет отвечать на все вопросы, и поэтому делаем так:

- 1) Если бот сильно уверен в ответе (score высокий), то даём пользователю ответ
- 2) Если бот совсем не уверен в ответе (score низкий), то переводим пользователя на оператора
- 3) Если score средний, то просим пользователя переформулировать вопрос.

Нужно выяснить, какой score считать низким, какой средним, а какой высоким. Мы провели эксперимент: попросили живых людей задать вопрос боту, а потом указать, что было бы правильно после такого вопроса сделать: ответить, переспросить или перевести на оператора. Мы записали в таблицу score от чёрного ящика и предложенное тестером действие. (table.csv)

Ваша задача - реализовать алгоритм, будет принимать на вход score и возвращать следующее действие: вывести ответ пользователю, переспросить пользователя или перевести на оператора. Обоснуйте, почему был выбран именно такой алгоритм и именно такие параметры для него.

Предложите метрику для определения точности своего алгоритма и посчитайте её значение для данных из table.csv

Формат решения: репозиторий на github/gitlab/bitbucket + google doc с обоснованием алгоритма и метрики.