Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Лабораторная работа №4

на тему

“Диалоговая система с поддержкой естественного языка”

по дисциплине «Естественно-языковой интерфейс интеллектуальных систем»

|  |  |
| --- | --- |
| Выполнил: | Колосовский Е. С.  Поплавский Э. Э.  Смелов А. А. |
| Группа: | 021703 |
| Проверил: | Name N. N. |

Минск 2023

Содержание

[1. Цель работы:](#_qgwal63jo2yb)

[2. Задачи лабораторной работы:](#_v2xm9djnywb7)

[3. Структурно-функциональная схема разработанного приложения.](#_z701oh6bkqde)

[4. Описание структур хранения данных, алгоритмов их обработки, необходимых для реализации базовых требований к разработанной программе.](#_ayvov9v3c3d6)

[5. Оценка быстродействия приложения.](#_qzit0noawv8k)

[6. Вывод.](#_h3i38hmtu3aj)

# Цель работы:

Освоить принципы разработки диалоговых систем с поддержкой естественного языка.

# Задачи лабораторной работы:

1. Изучить основы создания диалоговых систем с поддержкой естественного языка.
2. Закрепить навыки программирования при решении задач организации диалогового взаимодействия с поддержкой естественного языка.

Вариант лабораторной работы №22.

# Структурно-функциональная схема разработанного приложения.

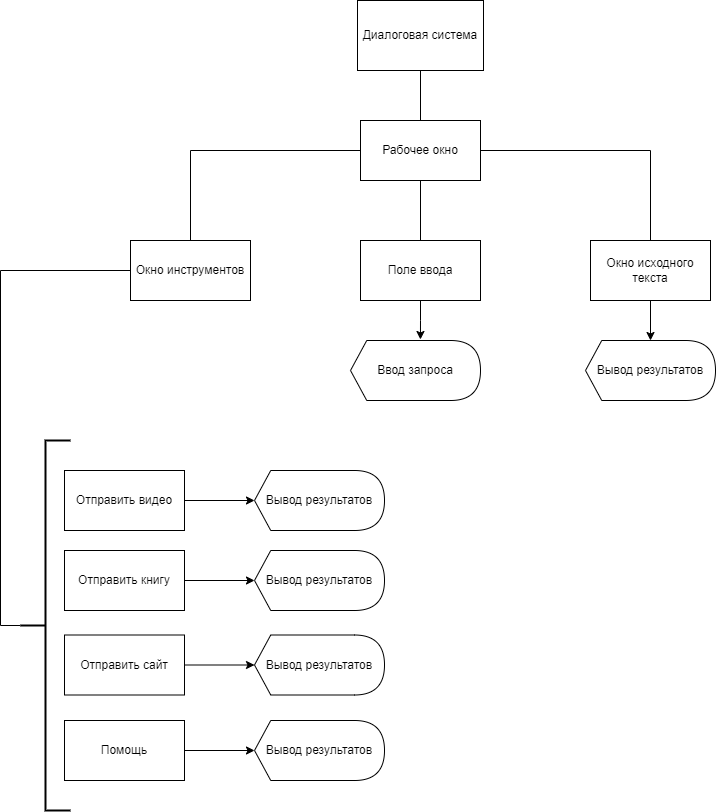


Рисунок 1. Структурно-функциональная схема разработанного приложения.

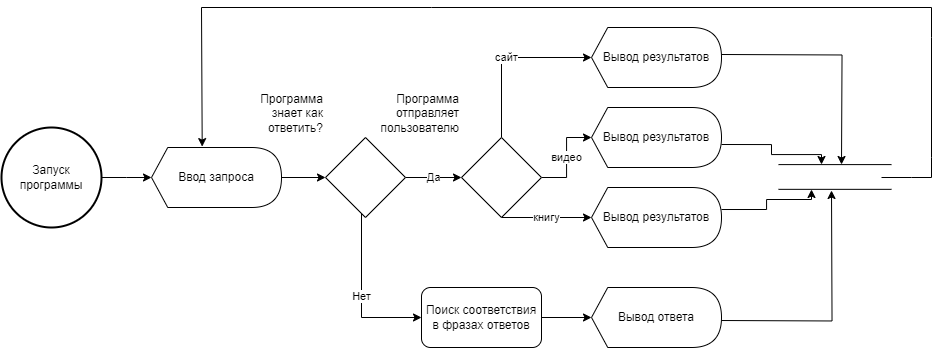


Рисунок 1. Принцип работы приложения.

# Описание структур хранения данных, алгоритмов их обработки, необходимых для реализации базовых требований к разработанной программе.

Для разработки диалогового помощника использовалась библиотека telebot для Python, которая позволяет создавать telegram-ботов. Скриншоты взаимодействия с ботом приведены ниже на рисунках 2-4.

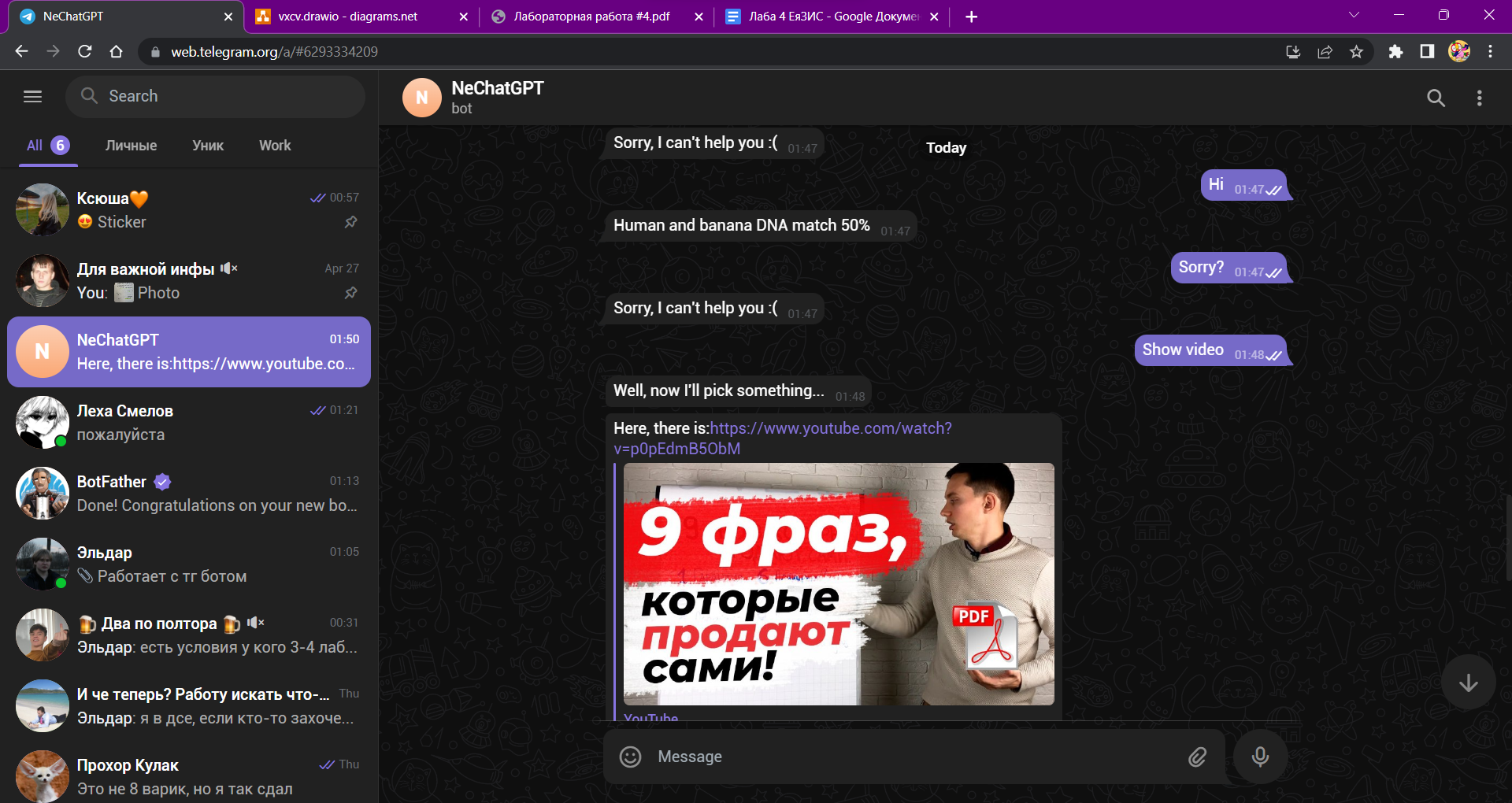


Рисунок 2. Пример диалога с ботом.

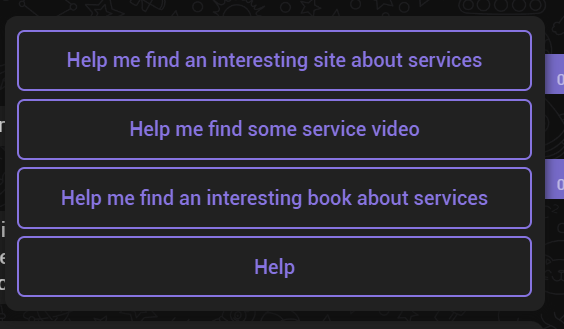


Рисунок 3. Доступные быстрые действия.

Фразы, которыми бот может отвечать хранятся в json-файле. Там заданы возможные запросы и соответственно ответы на них.

Для каждого запроса, наподобие отправь книгу или видео, предусмотрены свои функции, которые из уже заполненных массивов рандомно выбирают видео или книгу.

# Оценка быстродействия приложения.

В приложении все основные функции выполняются за время O(n). Только операция сортировки выходного словаря выполняется за время O(log n).

**O(n)** — линейная сложность.

Такой сложностью обладает, например, алгоритм поиска наибольшего элемента в не отсортированном массиве. Нам придётся пройтись по всем n элементам массива, чтобы понять, какой из них максимальный.

**O(n log n)** — логарифмическая сложность

Логарифмическая - O (log n). Означает, что сложность алгоритма растёт логарифмически с увеличением входных данных. Другими словами это такой алгоритм, где на каждой итерации берётся половина элементов. К алгоритмам с такой сложностью относятся алгоритмы типа “Разделяй и Властвуй” (Divide and Conquer), например бинарный поиск.

Используется Timsort (Рисунок 5) - стандартная сортировка языка Python.

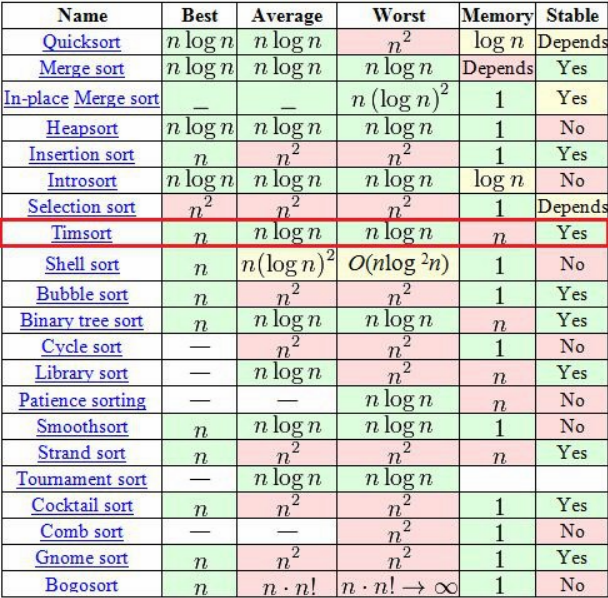


Рисунок 5. Сравнение быстродействия алгоритмов сортировки.

# [Вывод](#_h3i38hmtu3aj)

В ходе выполнения лабораторной работы были изучены назначение, структура и функциональность, предоставляемая базовым ЯП для разработки интеллектуального диалогового помощника.

Результатом лабораторной работы является telegram-бот, который позволяет получить информацию из сферы досуга, т.е. получить какое-то интересное видео для просмотра, книгу для прочтения или полезный сайт.