

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Студент	<u>Голикова С. М.</u>
Группа	<u>ИУ7-55Б</u>
Оценка (баллы)	<u> </u>
Преподаватели	Волкова Л. Л., Строганов Ю. В.

Москва, 2023г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитический раздел	4
1.1 Словарь	4
1.2 Алгоритм бинарного поиска	4
1.3 Лингвистическая переменная	5
1.4 Формализация объекта и его признаков	5
1.5 Требования к программному обеспечению	6
2 Конструкторский раздел	7
2.1 Метод поиска по словарю при ограничении на значение призна- ка, заданном при помощи лингвистической переменной	7
2.2 Разработка алгоритмов	7
3 Технологический раздел	13
3.1 Средства реализации	13
3.2 Представление данных	13
3.3 Реализации алгоритмов	14
3.4 Тестирование	18
4 Исследовательский раздел	19
4.1 Опрос респондентов	19
4.2 Демонстрация работы программы	20
ЗАКЛЮЧЕНИЕ	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24

ВВЕДЕНИЕ

Способность человека оценивать информацию играет существенную роль в определении сложных явлений [1]. В лабораторной работе рассматривается понятие лингвистической переменной, которая отличается от числовой тем, что ее значениями являются не числа, а слова или предложения в естественном или формальном языке. Поскольку слова в общем менее точны, чем числа, понятие лингвистической переменной дает возможность приближенно описывать явления, которые настолько сложны, что не поддаются описанию в общепринятых количественных терминах [2].

Словарь — это структура данных, позволяющая идентифицировать ее элементы не по числовому индексу, а по произвольному. Одной из основных операций над рассматриваемой структурой данных является поиск.

Целью данной работы является разработка метода поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

В рамках выполнения работы необходимо решить следующие задачи:

- формализовать объект и его признаки;
- провести анкетирование респондентов;
- описать алгоритм поиска в словаре объектов, удовлетворяющих ограничению, заданном в вопросе на ограниченном естественном языке;
- реализовать описанный алгоритм поиска в словаре;
- привести примеры запросов пользователя и сформированной реализацией алгоритма поиска выборки объектов из словаря, используя составленные респондентами вопросы;
- дать заключение о применимости предложенного алгоритма.

1 Аналитический раздел

В данном разделе приводятся определения словаря и лингвистической переменной, а также теоретическое описание алгоритма бинарного поиска. Кроме того, производится формализация объекта и его признаков.

1.1 Словарь

Словарь — структура данных, которая позволяет хранить пары вида «ключ — значение»: (k, v) . Ключ идентифицирует элемент словаря, значение является данными, которые соответствуют данному ключу. В словаре не может быть двух элементов с одинаковыми ключами, однако могут быть одинаковые значения у разных ключей [3].

1.2 Алгоритм бинарного поиска

Бинарный поиск производится в упорядоченном массиве. При бинарном поиске искомый ключ сравнивается с ключом среднего элемента в массиве. Если они равны, то поиск успешен. В противном случае поиск осуществляется аналогично в левой или правой частях массива [4].

Пусть n — количество элементов в массиве. Тогда сложность алгоритма в худшем и среднем случае равна $O(\log n)$, в лучшем — $O(1)$.

У бинарного поиска есть недостаток: он требует упорядочивания данных по возрастанию. Сложность сортировки — не менее $O(n \cdot \log n)$. Поэтому бинарный поиск чаще применяется на массивах большого размера.

1.3 Лингвистическая переменная

Лингвистическая переменная характеризуется набором свойств $(X, T(X), U, G, M)$, в котором X — название переменной; $T(X)$ — терм-множество переменной X , т.е. множество названий лингвистических значений переменной X , причем каждое из таких значений является нечеткой переменной \tilde{x} со значениями из универсального множества U с базовой переменной u ; G — синтаксическое правило, порождающее названия \tilde{x} значений переменной X ; M — семантическое правило, которое ставит в соответствие каждой нечеткой переменной \tilde{x} ее смысл $M(\tilde{x})$, т.е. нечеткое подмножество $M(\tilde{x})$ универсального множества U .

Конкретное название \tilde{x} , порожденное синтаксическим правилом G , называется термом. Терм, который состоит из одного слова или из нескольких слов, всегда фигурирующих вместе друг с другом, называется атомарным термом. Терм, который состоит из более чем одного атомарного термина, называется составным термом [2].

1.4 Формализация объекта и его признаков

Рассмотрим лингвистическую переменную с именем $X = \text{«Планеты Солнечной системы»}$.

Универсальное множество $U = [\text{«Меркурий»}, \text{«Венера»}, \text{«Марс»}, \text{«Юпитер»}, \text{«Сатурн»}, \text{«Уран»}, \text{«Нептун»}]$.

Терм-множество $T = [\text{«Близко»}, \text{«Средне»}, \text{«Далеко»}]$. Данные термы показывают дальность расположения конкретной планеты относительно Земли.

Синтаксическое правило G порождает новые термы с использованием квантификаторов «не» и «очень».

M — процедура, ставящая каждому новому терму в соответствие нечеткое множество из X по правилам: если терм t имел функцию принадлежности $\mu_t(u)$, то новые термы будут иметь функции принадлежности, заданные в таблице 1.

Таблица 1 — Функции принадлежности составных термов

Квантификатор	Функция принадлежности
не t	$1 - \mu_t(u)$
очень t	$(\mu_t(u))^2$

1.5 Требования к программному обеспечению

Программа должна предоставлять следующие возможности:

- ввод запроса;
- вывод на экран найденных в словаре объектов, удовлетворяющих ограничению, заданному в вопросе.

Вывод

Были рассмотрены идеи, необходимые для разработки и реализации алгоритма бинарного поиска, а также определения терминов, используемых в лабораторной работе.

2 Конструкторский раздел

В данном разделе описывается метод поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной, а также приводятся разработанные алгоритмы бинарного поиска и получения объектов по заданному в вопросе на естественном языке ограничению.

2.1 Метод поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной

Этапы метода поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной:

- 1) выполнить графематический анализ входного вопроса — предложения на русском языке;
- 2) выполнить морфологический анализ слов из вопроса;
- 3) проверить наличие в вопросе указания на объект — планету — и на Землю;
- 4) выделить терм и квантификаторы;
- 5) сформировать ограничение на диапазон значений расстояния соответственно выделенным терму и квантификаторам;
- 6) выполнить поиск в словаре соответственно сформированным ограничениям;
- 7) подготовить результат к выводу в виде текста на естественном языке.

2.2 Разработка алгоритмов

На рисунке 1 приведена схема алгоритма бинарного поиска в словаре. На рисунке 2 приведены схемы алгоритмов получения объекта по значению функ-

ции принадлежности, а также расчета функций принадлежности для квантификаторов «не» и «очень». На рисунках 3–4 приведена схема алгоритма получения объекта по заданному в вопросе на естественном языке ограничению.

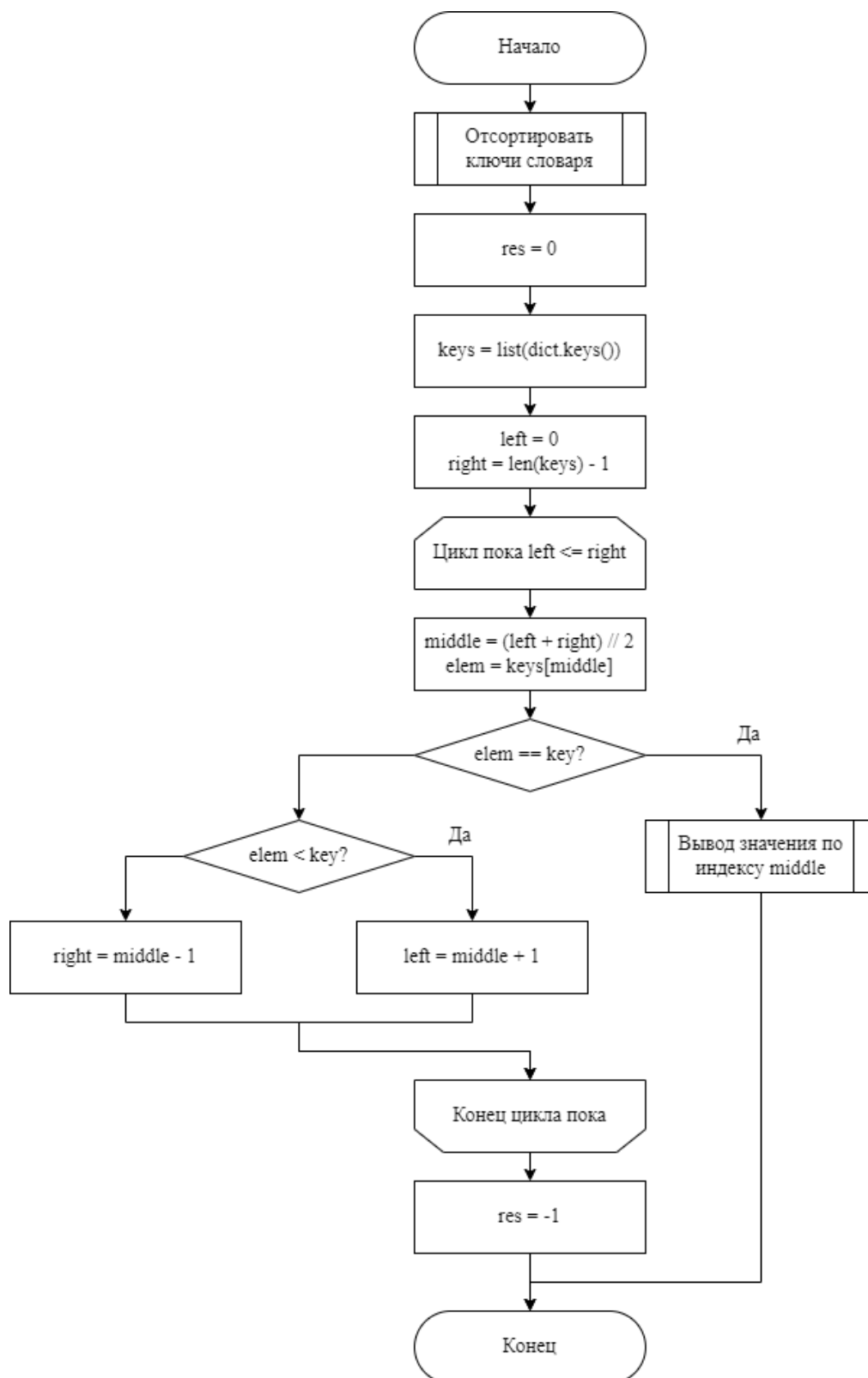


Рисунок 1 — Алгоритм бинарного поиска

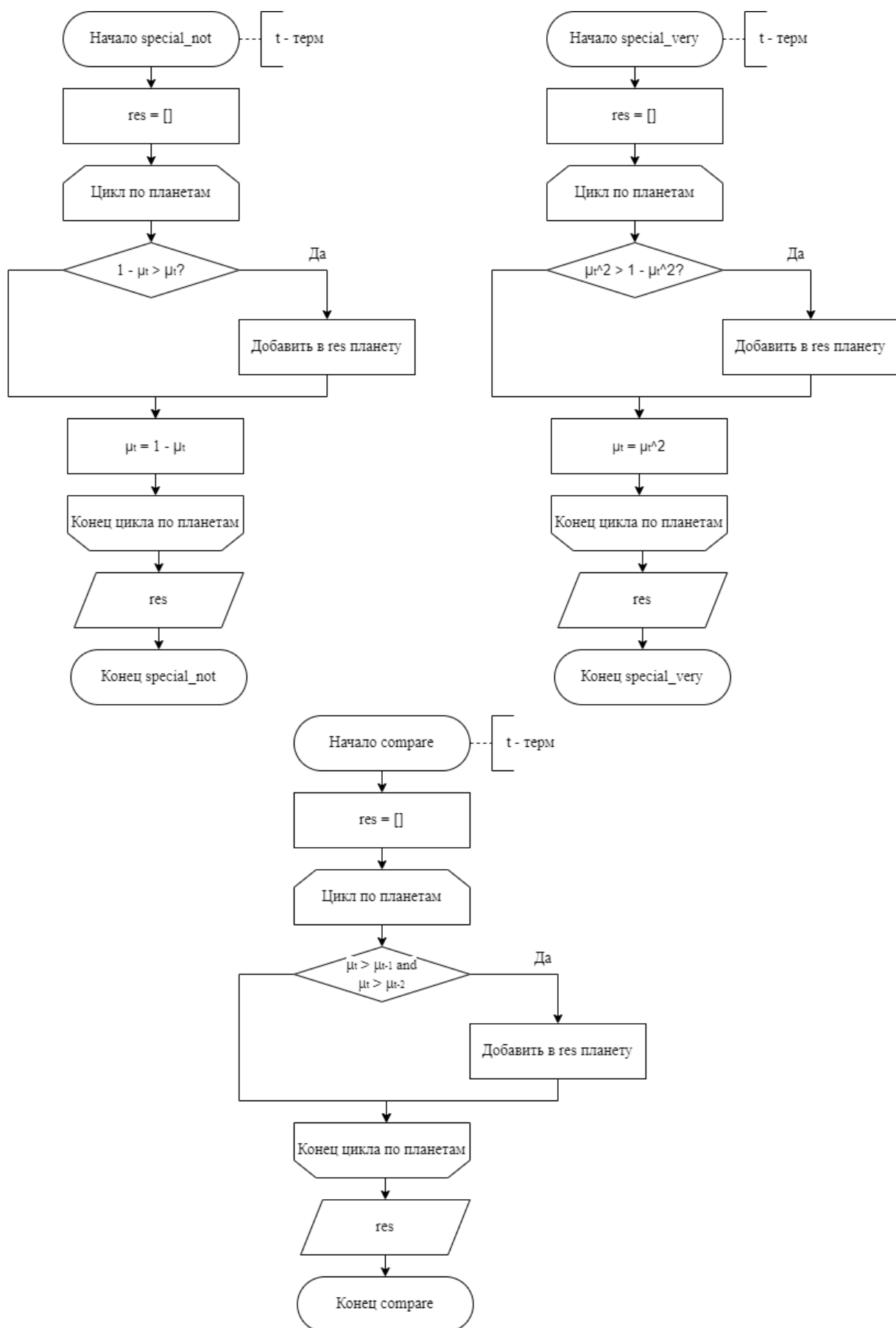


Рисунок 2 — Алгоритмы получения объектов по значениям функций принадлежности

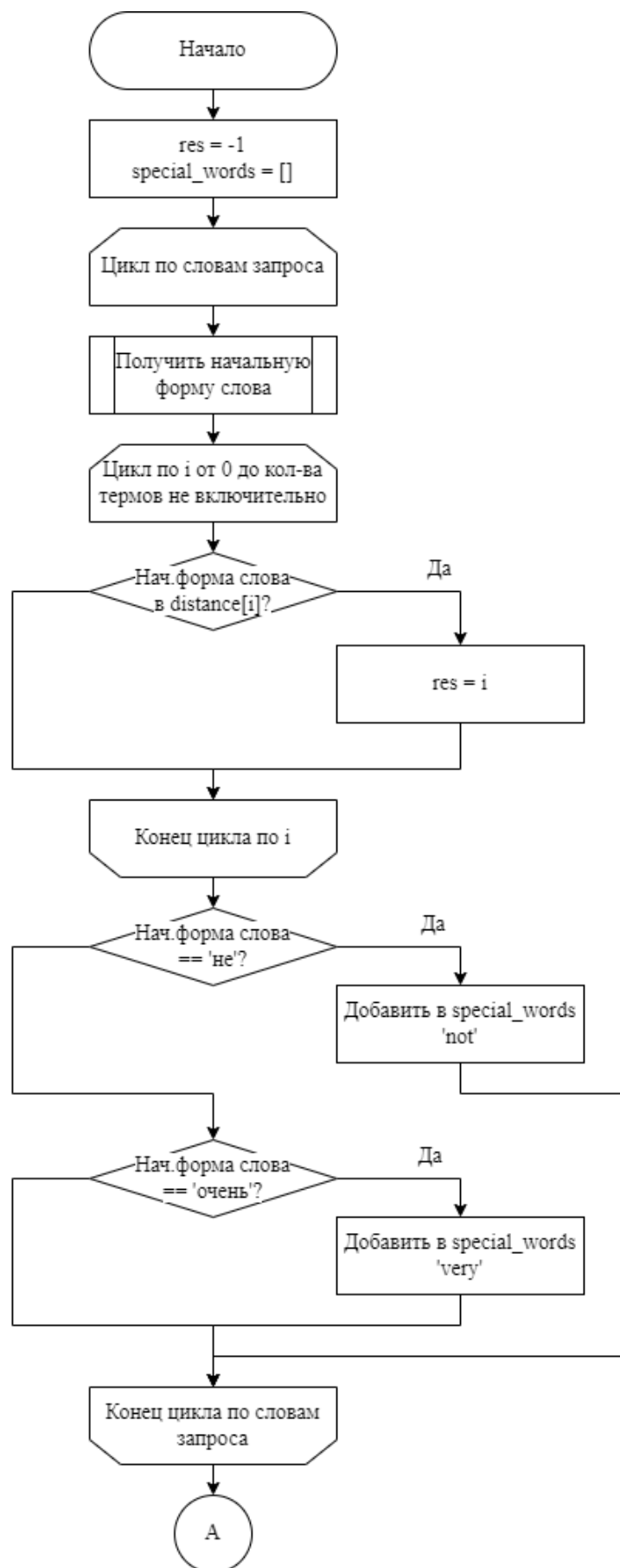


Рисунок 3 — Алгоритм получения объектов по ограничению

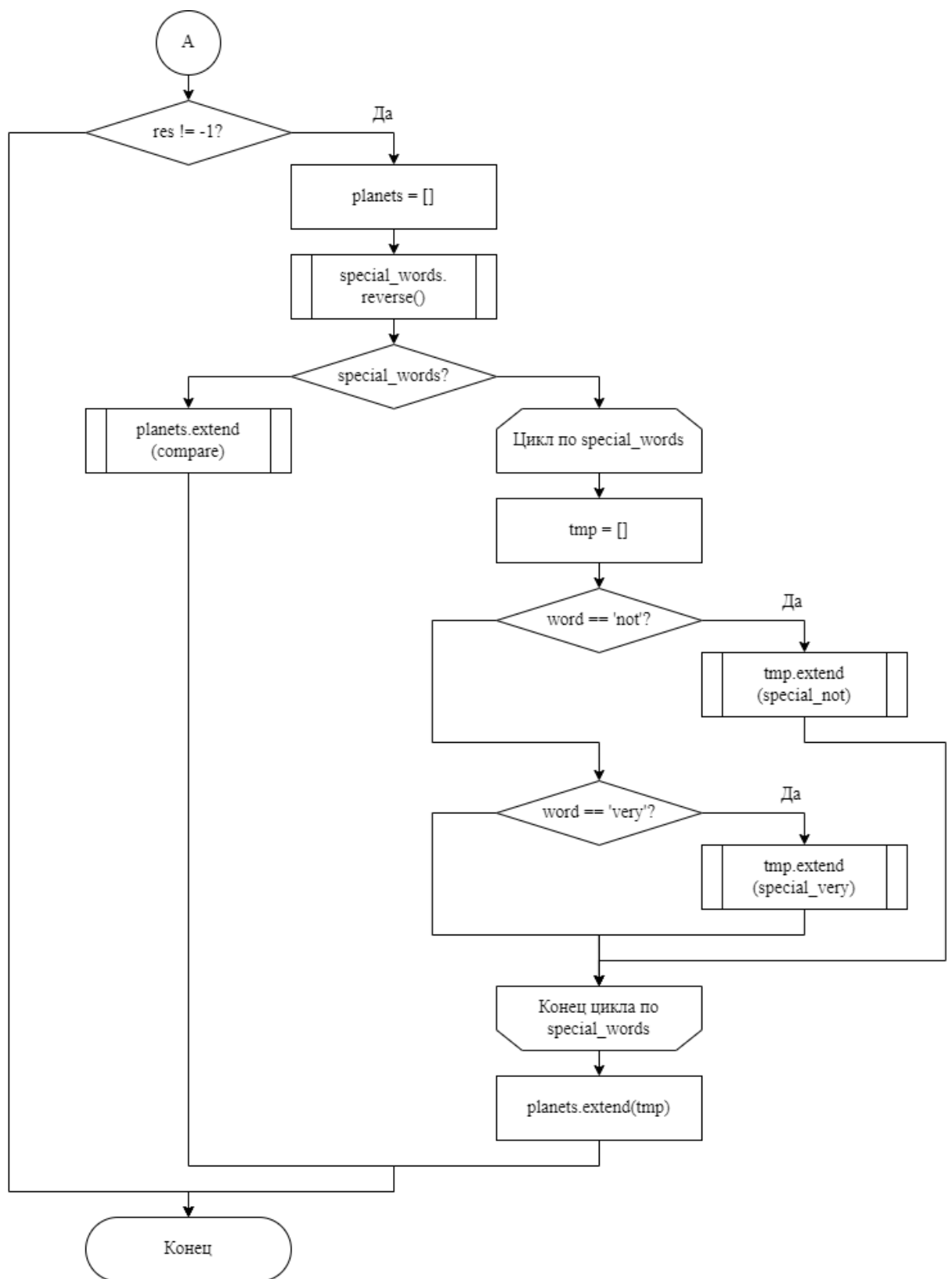


Рисунок 4 — Алгоритм получения объектов по ограничению (продолжение)

Вывод

Были разработаны алгоритмы бинарного поиска и получения объектов по заданному в вопросе на естественном языке ограничению.

3 Технологический раздел

В данном разделе приводятся описание средств реализации и сами реализации алгоритмов, а также функциональные тесты.

3.1 Средства реализации

В качестве языка программирования для реализации лабораторной работы был выбран Python [5]. Данный язык предоставляет возможности работы с массивами и словарями. Кроме того, в языке есть морфологический анализатор `rumorphy2`, позволяющий получать начальные формы слов.

3.2 Представление данных

В реализации алгоритма получения объектов, удовлетворяющих заданному ограничению, используются переменные `distance` и `planets_dist`.

`distance` представляет собой массив из 3 массивов, каждый из которых содержит синонимы конкретного термина («близко», «средне» или «далеко»).

`planets_dist` является словарем, в котором ключами выступают планеты, а значениями — массивы из 3 элементов, содержащие значения функций принадлежности для каждого термина.

Сам словарь, в котором осуществляется поиск, является аналогией обычного бумажного словаря, в котором в качестве ключа выступает слово, а в качестве значения — страница. Слова и страницы генерируются случайным образом, но среди слов обязательно есть названия всех планет Солнечной системы.

3.3 Реализации алгоритмов

В листинге 1 приведена реализация алгоритма бинарного поиска в словаре. В листинге 2 — реализация алгоритмов получения объектов по значению функции принадлежности, а также вычисления значений этой функции для квантификаторов «не» и «очень». В листинге 3 — реализация алгоритма получения объектов по заданному на естественном языке ограничению.

Листинг 1 — Реализация алгоритма бинарного поиска

```
1 def sort_dict(my_dict):
2     keys = list(my_dict.keys())
3     keys.sort()
4     tmp_dict = dict()
5     for key in keys:
6         tmp_dict[key] = my_dict[key]
7
8     return tmp_dict
9
10 def binary_search(sort_dict, find_key, output=True):
11     result = -1
12     keys = list(sort_dict.keys())
13     left, middle, right = 0, len(keys) // 2, len(keys) - 1
14     while left <= right:
15         key = keys[middle]
16         if key == find_key:
17             if output:
18                 print(list(sort_dict.items())[middle])
19                 result = 0
20                 break
21             elif key < find_key:
22                 left = middle + 1
23             else:
24                 right = middle - 1
25             middle = (left + right) // 2
26
27     return result
28
29 def process_binary_search(global_dict, find_key):
30     sorted_dict = sort_dict(global_dict)
31     result = binary_search(sorted_dict, find_key)
32     if result == -1:
33         print("Key not found")
```

Листинг 2 — Реализация алгоритмов вычисления значений функций принадлежности

```
1 def get_planets_by_dist(pl_dist, t):
2     res = []
3     for planet in pl_dist.keys():
4         if pl_dist[planet][t] > pl_dist[planet][t - 1] and \
5             pl_dist[planet][t] > pl_dist[planet][t - 2]:
6             res.append(planet)
7
8     return res
9
10 def inv(a):
11     return 1 - a
12
13 def special_not(pl_dist, t):
14     res = []
15     for planet in pl_dist.keys():
16         if inv(pl_dist[planet][t]) > pl_dist[planet][t]:
17             res.append(planet)
18         pl_dist[planet][t] = inv(pl_dist[planet][t])
19
20     return res
21
22 def special_very(pl_dist, t):
23     res = []
24     for planet in pl_dist.keys():
25         if (pl_dist[planet][t])**2 > inv((pl_dist[planet][t])**2):
26             res.append(planet)
27         pl_dist[planet][t] = (pl_dist[planet][t]) ** 2
28
29     return res
```


Листинг 3 — Реализация алгоритма получения удовлетворяющих заданному ограничению объектов

```
1 def get_planets(question_split, distance, planets_dist):
2     res = -1
3     special_words = []
4
5     for word in question_split:
6         n_form = morph.parse(word)[0].normal_form
7         for i in range(3):
8             if n_form in distance[i]:
9                 res = i
10
11         if n_form == not_word:
12             special_words.append('not')
13
14         if n_form == very_word:
15             special_words.append('very')
16
17     if res == -1:
18         print("\nThe question is not recognized :(")
19     else:
20         planets = []
21         special_words.reverse()
22
23         if not special_words:
24             planets.extend(get_planets_by_dist(planets_dist, res))
25         else:
26             tmp = []
27             for i in special_words:
28                 tmp = []
29                 if i == 'not':
30                     tmp.extend(special_not(planets_dist, res))
31                 elif i == 'very':
32                     tmp.extend(special_very(planets_dist, res))
33             planets.extend(tmp)
```

3.4 Тестирование

В таблице 2 приведены функциональные тесты. Тесты были успешно пройдены программой.

Таблица 2 — Функциональное тестирование

Запрос	Ожидаемый ответ
Какие планеты находятся рядом с Землей?	Венера, Марс
Планеты на среднем расстоянии от Земли	Меркурий, Юпитер
планеты очень далеко от земли	Уран, Нептун
планеты не далеко от земли	Меркурий, Венера, Марс, Юпитер
планеты очень не далеко от земли	Венера, Марс
планеты не очень далеко от земли	Меркурий, Венера, Марс, Юпитер, Сатурн
самая дальняя звезда	Вопрос не распознан :(
борщ рецепт	Вопрос не распознан :(

Вывод

Были описаны средства реализации, приведены реализации алгоритмов и тесты, успешно пройденные программой.

4 Исследовательский раздел

В данном разделе приводятся результаты опроса респондентов и примеры работы программы.

4.1 Опрос респондентов

В опросе принимали участие следующие студенты:

- 1) Григоренко Е. А. (ИУ7-55Б);
- 2) Иванов П. А. (ИУ7-55Б);
- 3) Ильин Д. А. (ИУ7-55Б);
- 4) Киселева М. С. (ИУ7-56Б);
- 5) Саркисян А. А. (ИУ7-55Б).

Результаты опроса приведены в таблице 3, где Б — близко, С — средне, Д — далеко относительно Земли.

Таблица 3 — Результаты опроса

Респондент	Меркурий	Венера	Марс	Юпитер	Сатурн	Уран	Нептун
1	С	Б	Б	Д	Д	Д	Д
2	Д	Б	Б	Д	Д	Д	Д
3	С	Б	Б	С	С	Д	Д
4	Д	Б	С	С	С	Д	Д
5	С	Б	Б	С	Д	Д	Д

Количество людей, выбравших тот или иной вариант для конкретной планеты, представлено в таблице 4.

Таблица 4 — Сводная таблица

	Меркурий	Венера	Марс	Юпитер	Сатурн	Уран	Нептун
Б	0	5	4	0	0	0	0
С	3	0	1	3	2	0	0
Д	2	0	0	2	3	5	5

На рисунке 5 представлены графики функций принадлежности значений переменной термам, описывающим группы значений лингвистической переменной.

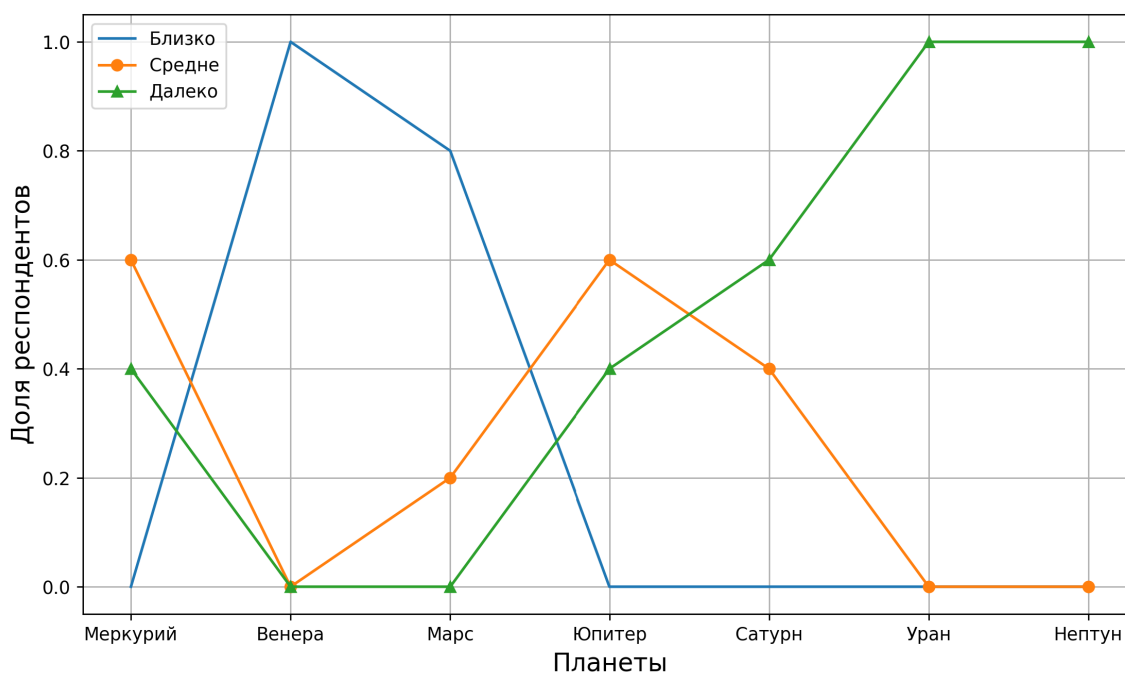


Рисунок 5 — Функции принадлежности термам значений переменной

4.2 Демонстрация работы программы

На рисунке 9 представлен пример работы программы. Пользователь вводит вопрос на естественном языке, программа выдает результаты в виде кортежей («слово», «страница в словаре»).

Введите запрос: *Какие планеты рядом с Землей?*

Результаты:

('Венера', '166')

('Марс', '360')

Рисунок 6 — Пример работы реализаций алгоритмов

Введите запрос: *планеты очень далеко от земли*

Результаты:

('Уран', '706')

('Нептун', '777')

Рисунок 7 — Пример работы реализаций алгоритмов

Введите запрос: *планеты не очень далеко от земли*

Результаты:

('Меркурий', '431')

('Венера', '166')

('Марс', '360')

('Юпитер', '221')

('Сатурн', '511')

Рисунок 8 — Пример работы реализаций алгоритмов

Введите запрос: *самая яркая звезда*

Вопрос не распознан :(

Рисунок 9 — Пример работы реализаций алгоритмов

Вывод

Были приведены результаты опроса респондентов и построены графики функций принадлежности значений переменной термам. Также были рассмотрены примеры работы программы.

Разработанный метод рекомендуется к применению. Ограничением метода является соответствие круга респондентов кругу пользователей: если бы респондентами выступали астрономы, то потребовалось бы провести отдельный опрос среди соответствующей профильной выборки респондентов, что дало бы иные результаты в части диапазонов значений расстояний между небесными телами, соответствующих выбранным термам.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были решены следующие задачи:

- формализованы объект и его признаки;
- проведено анкетирование респондентов;
- описан алгоритм бинарного поиска в словаре объектов, удовлетворяющих ограничению, заданном в вопросе на ограниченном естественном языке;
- реализован описанный алгоритм бинарного поиска в словаре;
- приведены примеры запросов пользователя и сформированной реализацией алгоритма поиска выборки объектов из словаря;
- дано заключение о применимости предложенного алгоритма.

Поставленная цель была достигнута: разработан метод поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

Даны рекомендации о применимости предложенного метода, выделены ограничения его применимости.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Нечеткие и лингвистические переменные [Электронный ресурс]. Режим доступа: <http://dit.isuct.ru/IVT/BOOKS/IS/Model/gl43.htm> (дата обращения: 18.12.2022).
2. Лингвистическая нечеткая логика [Электронный ресурс]. Режим доступа: <https://intuit.ru/studies/courses/87/87/lecture/20515> (дата обращения: 18.12.2022).
3. Словари [Электронный ресурс]. Режим доступа: <https://younglinux.info/python/dictionary> (дата обращения: 18.12.2022).
4. Бинарный поиск [Электронный ресурс]. Режим доступа: <https://prog-cpp.ru/search-binary/> (дата обращения: 18.12.2022).
5. Лутц, Марк. Изучаем Python, том 1, 5-е изд. — Пер. с англ. — СПб.: ООО “Диалектика”, 2019. — 832 с.