



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №6

Название: Поиск по словарю

Дисциплина: Анализ алгоритмов

Студент ИУ7-52Б
(Группа)

(Подпись, дата) Н. В. Ляпина
(И.О. Фамилия)

Преподаватель

(Подпись, дата) Л.Л. Волкова
(И.О. Фамилия)

Москва, 2022

Содержание

Введение	3
1 Аналитический раздел	4
1.1 Нечеткие переменные	4
1.2 Анкетирование	4
1.3 Формализация объекта	5
1.4 Словарь как структура данных	5
1.5 Алгоритм полного перебора	6
2 Конструкторский раздел	8
2.1 Структура анкеты	8
2.2 Типовые запросы	8
2.3 Описание используемых типов данных	8
2.4 Структура ПО	8
2.5 Схема алгоритма	9
3 Технологический раздел	11
3.1 Результаты анкетирования	11
3.2 Функция принадлежности	12
3.3 Требования к ПО	14
3.4 Средства реализации	14
3.5 Листинг программы	14
3.6 Тестирование ПО	16
Заключение	19
Список источников	20

Введение

Словарь, как тип данных, применяется везде, где есть связь «ключ – значение» или «объект – данные»: поиск налогов по ИНН и другое. Поиск - основная задача при использовании словаря. Но также важно, правильно задать вопрос, чтобы поисковая система в словаре могла выдать однозначный результат.

Цель лабораторной работы:

получить навык поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной.

Задачи лабораторной работы:

- 1) формализовать объект и его признак;
- 2) составить анкету для её заполнения респондентом;
- 3) провести анкетирование респондентов;
- 4) построить функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов, выступающих в роли экспертов;
- 5) описать 3-5 типовых вопросов на русском языке, имеющих целью запрос на поиск в словаре;
- 6) описать алгоритм поиска в словаре объектов, удовлетворяющих ограничению, заданному в вопросе на ограниченном естественном языке;
- 7) описать структуру данных словаря, хранящего наименования объектов согласно варианту и числовое значение признака объекта;
- 8) реализовать алгоритм поиска в словаре;
- 9) привести примеры запросов пользователя и сформированный реализацией алгоритм поиска выборки объектов из словаря, используя составленные респондентами вопросы;
- 10) дать заключение о применимости предложенного алгоритма и о его ограничениях.

1 Аналитический раздел

1.1 Нечеткие переменные

Лингвистической переменной являются слова или предложения естественного языка, которые описываются нечеткими значениями.

В случаях, когда требуется формализовать разделение диапазона значений данной величины на некоторые категории следует выделить набор категорий и определить границы диапазонов значений в пределах каждой категории. Для этого требуется агрегировать набор значений и интерпретировать набор мнений экспертов касательно соотношения категории и конкретных значений описываемой величины [1].

При построении функции принадлежности значений категориям используются два множества:

- 1) множество термов τ , которое описывает лингвистических терм $\tau'_i = (t_1, t_2, \dots, t_m)$, $i \in [1, m]$ на универсальном множестве $X = (x_1, x_2, \dots, x_n)$
- 2) собственно само универсальное множество $X = (x_1, x_2, \dots, x_n)$

Термом являются слова или предложения естественного языка, которые описываются нечеткими значениями.

Универсальное множество – рассматриваемое множество значений.

$$\tau'_i = \left(\frac{\mu_{\tau_i}(x_1)}{x_1}, \frac{\mu_{\tau_i}(x_2)}{x_2}, \dots, \frac{\mu_{\tau_i}(x_n)}{x_n} \right) \quad (1.1)$$

Требуется определить для любых $i \in [1, m], j \in [1, n]$ степени функций принадлежности элементов множества X к элементам из множества τ'_i , т.е. $\mu_{\tau_i}(x_j)$.

Для решения данной задачи будет использовать метод на основе статистической обработки мнений экспертов (респондентов).

1.2 Анкетирование

В данном методе усредняют знания коллектива специалистов относительно распределения экспертов по универсальному множеству. Анкета будет иметь следующий вид:

где k – количество экспертов, $a_{nm}^k \in [0, 1]$ – результат бинарной экспертной оценки k -м экспертов у элемента x_n свойств нечеткого множества $\tau'_i, j \in [1, n], i \in [1, m], k \in [1, p]$.

По результатам анкетирования рассчитываются степени функции принадлежности нечетному множеству:

$$\tau'_i : \mu_{\tau'_i}(x_n) = \frac{1}{k} \sum_{k=1}^k a_{nm}^k \quad (1.2)$$

Таблица 1.1 — Анкета для k экспертов

		x_1	x_2	\dots	x_n
τ'_1	t_1	a_{11}^1	a_{21}^1	\dots	a_{n1}^1
	t_2	a_{12}^1	a_{22}^1	\dots	a_{n2}^1
	\dots	\dots	\dots	\dots	\dots
	t_m	a_{1m}^1	a_{2m}^1	\dots	a_{nm}^1
τ'_2	t_1	a_{11}^2	a_{21}^2	\dots	a_{n1}^2
	t_2	a_{12}^2	a_{22}^2	\dots	a_{n2}^2
	\dots	\dots	\dots	\dots	\dots
	t_m	a_{1m}^2	a_{2m}^2	\dots	a_{nm}^2
\dots	\dots	\dots	\dots	\dots	\dots
τ'_k	t_1	a_{11}^k	a_{21}^k	\dots	a_{n1}^k
	t_2	a_{12}^k	a_{22}^k	\dots	a_{n2}^k
	\dots	\dots	\dots	\dots	\dots
	t_m	a_{1m}^k	a_{2m}^k	\dots	a_{nm}^k

1.3 Формализация объекта

В качестве лингвистической переменной в данной работе будет рассмотрен размер отчета по Анализу алгоритмов. Она принимает нечеткие значения:

- 1) «очень маленький»;
- 2) «маленький»;
- 3) «небольшой»;
- 4) «средний»;
- 5) «большой»;
- 6) «очень большой».

Эти нечеткие значения образуют множество термов.

Данное множество термов будет рассмотрено на универсальном множестве $X = [0,50]$.

1.4 Словарь как структура данных

Словарь (или «ассоциативный массив») 2 - абстрактный тип данных (интерфейс к хранилищу данных), позволяющий хранить пары вида «(ключ, значение)» и поддерживающий операции добавления пары, а также поиска и удаления пары по ключу:

- INSERT(k, v);
- FIND(k);
- REMOVE(k).

В паре (k, v) : v называется значением, ассоциированным с ключом k . Где k — это ключ, а v — значение. Семантика и названия вышеупомянутых операций в разных реализациях ассоциативного массива могут отличаться.

Операция $\text{ПОИСК}(k)$ возвращает значение, ассоциированное с заданным ключом, или некоторый специальный объект НЕ_НАЙДЕНО , означающий, что значения, ассоциированного с заданным ключом, нет. Две другие операции ничего не возвращают (за исключением, возможно, информации о том, успешно ли была выполнена данная операция).

Ассоциативный массив с точки зрения интерфейса удобно рассматривать как обычный массив, в котором в качестве индексов можно использовать не только целые числа, но и значения других типов — например, строки.

В данной лабораторной работе в качестве ключа будет использоваться строка: название отчета, а в качестве значения — целое число: количество страниц в данном отчете.

1.5 Алгоритм полного перебора

Алгоритмом полного перебора \mathcal{Z} называют метод решения задачи, при котором по очереди рассматриваются все возможные варианты. В случае реализации алгоритма в рамках данной работы будут последовательно перебираться ключи словаря до тех пор, пока не будет найден нужный.

Трудоёмкость алгоритма зависит от того, присутствует ли искомый ключ в словаре, и, если присутствует — насколько он далеко от начала массива ключей.

Пусть на старте алгоритм затрагивает k_0 операций, а при сравнении k_1 операций.

Пусть алгоритм нашёл элемент на первом сравнении (лучший случай), тогда будет затрачено $k_0 + k_1$ операций, на втором — $k_0 + 2 \cdot k_1$, на последнем (худший случай) — $k_0 + N \cdot k_1$. Если ключа нет в массиве ключей, то мы сможем понять это, только перебрав все ключи, таким образом трудоёмкость такого случая равно трудоёмкости случая с ключом на последней позиции. Средняя трудоёмкость может быть рассчитана как математическое ожидание по формуле (1.3), где Ω — множество всех возможных случаев.

$$\begin{aligned}
 \sum_{i \in \Omega} p_i \cdot f_i &= (k_0 + k_1) \cdot \frac{1}{N+1} + (k_0 + 2 \cdot k_1) \cdot \frac{1}{N+1} + \\
 &+ (k_0 + 3 \cdot k_1) \cdot \frac{1}{N+1} + (k_0 + N k_1) \frac{1}{N+1} + (k_0 + N \cdot k_1) \cdot \frac{1}{N+1} = \\
 &= k_0 \frac{N+1}{N+1} + k_1 + \frac{1+2+\dots+N+N}{N+1} = \\
 &= k_0 + k_1 \cdot \left(\frac{N}{N+1} + \frac{N}{2} \right) = k_0 + k_1 \cdot \left(1 + \frac{N}{2} - \frac{1}{N+1} \right)
 \end{aligned} \tag{1.3}$$

Вывод

Программное обеспечение, решающее поставленную задачу, может работать следующим образом. На вход алгоритму подается словарь, содержащий данные об отчетах и их раз-

мерах, а также запрос к этому словарю. Программа возвращает массив элементов, которые удовлетворяют данному запросу.

2 Конструкторский раздел

В данном разделе будет составлена анкета для ее заполнения респондентами, описаны типовые запросы; будет рассмотрена схема алгоритма и требования к функциональности ПО.

2.1 Структура анкеты

В таблице 2.1 представлена анкета для заполнения респондентами. Респондент должен ответить утверждением или нет для каждого из значений данной анкеты.

Таблица 2.1 — Анкета для заполнения респондентами

	0	5	10	15	20	25	30	35	40	45	50
очень маленький											
маленький											
небольшой											
средний											
большой											
очень большой											

2.2 Типовые запросы

Таким образом типовые запросы к словарю выглядят так:

- «Вывести все небольшие отчеты»;
- «Найти большие и очень большие отчеты»;
- «Маленькие отчеты»;
- «Показать средние и небольшие отчеты».

Из типовых запросов видно, что запросы регистронезависимы и могут использовать более одного термина.

2.3 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

- словарь — встроенный тип `dict` [4] в Python[5] будет использован в качестве словаря;
- массив с результатами запроса — встроенный тип `list` [6] в Python.

2.4 Структура ПО

В данном ПО будет реализован метод структурного программирования. Взаимодействие с пользователем будет через консоль, будет дана возможность ввода запроса для поиска значений в словаре. Для работы будут разработаны следующие процедуры:

- главная процедура – является точкой входа в программу, входных данных нет, выходных данных нет;
- процедура обработки введенного запроса – анализирует строку с запросом и вычленяет из нее разделители и термы для поиска;
- процедура поиска – исходя из полученных массивов разделителей и термов возвращает массив результатов;
- процедура печати – получает массив результатов и построчно выводит элементы массива на экран.

2.5 Схема алгоритма

На рисунке 2.1 представлена схема алгоритма поиска в словаре полным перебором.



Рисунок 2.1 — Функция принадлежности для лингвистической переменной «размер отчета по
Аналізу алгоритмов»

Вывод

Были разработаны схемы алгоритмов, необходимых для решения задачи. Получено достаточно теоретической информации для написания программного обеспечения.

3 Технологический раздел

В данном разделе представлены результаты анкетирования, построена функция принадлежности, представлены средства, использованные в процессе разработки для реализации задачи, а также листинг кода программы. Кроме того показаны результаты тестирования.

3.1 Результаты анкетирования

В таблице 3.1 представлены результаты анкетирования 5-ти респондентов. Заполнение анкет выполнялось по бинарному признаку: «1» ставилась там, где утверждение считалось респондентами верным, и «0» там, где нет соответственно.

Таблица 3.1 — Результаты анкетирования

		0	5	10	15	20	25	30	35	40	45	50
Карина	очень маленький	1	1	0	0	0	0	0	0	0	0	0
	маленький	0	0	1	1	0	0	0	0	0	0	0
	небольшой	0	0	0	0	1	1	0	0	0	0	0
	средний	0	0	0	0	0	0	1	1	0	0	0
	большой	0	0	0	0	0	0	0	0	1	1	0
	очень большой	0	0	0	0	0	0	0	0	0	0	1
Вероника	очень маленький	1	1	0	0	0	0	0	0	0	0	0
	маленький	0	0	1	0	0	0	0	0	0	0	0
	небольшой	0	0	0	1	0	0	0	0	0	0	0
	средний	0	0	0	0	1	1	1	0	0	0	0
	большой	0	0	0	0	0	0	0	1	1	1	0
	очень большой	0	0	0	0	0	0	0	0	0	0	1
Илья	очень маленький	1	1	0	0	0	0	0	0	0	0	0
	маленький	0	0	1	1	0	0	0	0	0	0	0
	небольшой	0	0	0	0	1	0	0	0	0	0	0
	средний	0	0	0	0	0	1	1	0	0	0	0
	большой	0	0	0	0	0	0	0	1	1	1	0
	очень большой	0	0	0	0	0	0	0	0	0	0	1
Леша	очень маленький	1	1	0	0	0	0	0	0	0	0	0
	маленький	0	1	1	0	0	0	0	0	0	0	0
	небольшой	0	0	1	1	0	0	0	0	0	0	0
	средний	0	0	0	1	1	1	1	0	0	0	0
	большой	0	0	0	0	0	0	1	1	0	0	0
	очень большой	0	0	0	0	0	0	0	0	1	1	1
Катя	очень маленький	1	1	1	0	0	0	0	0	0	0	0
	маленький	0	0	0	1	0	0	0	0	0	0	0
	небольшой	0	0	0	0	1	0	0	0	0	0	0
	средний	0	0	0	0	0	1	0	0	0	0	0
	большой	0	0	0	0	0	0	1	1	0	0	0
	очень большой	0	0	0	0	0	0	0	0	1	1	1

3.2 Функция принадлежности

Благодаря полученным результатам была построена таблица 3.2, где приведена сумма голосов респондентов для каждого терма τ'_i , а также $\mu_i(x_n)$ согласно формуле 1.2.

Таблица 3.2 — Суммированные результаты анкетирования

	0	5	10	15	20	25	30	35	40	45	50	
очень маленький	5	5	1	0	0	0	0	0	0	0	0	Sum
	1	1	0.2	0	0	0	0	0	0	0	0	mu
маленький	0	1	4	3	0	0	0	0	0	0	0	Sum
	0	0.2	0.8	0.6	0	0	0	0	0	0	0	mu
небольшой	0	0	1	2	3	1	0	0	0	0	0	Sum
	0	0	0.2	0.4	0.6	0.2	0	0	0	0	0	mu
средний	0	0	0	1	2	4	4	1	0	0	0	Sum
	0	0	0	0.2	0.4	0.8	0.8	0.2	0	0	0	mu
большой	0	0	0	0	0	0	2	4	3	3	0	Sum
	0	0	0	0	0	0	0.4	0.8	0.6	0.6	0	mu
очень большой	0	0	0	0	0	0	0	0	2	2	5	Sum
	0	0	0	0	0	0	0	0	0.4	0.4	1	mu

Используя информацию из полученной таблицы, был построен график функции принадлежности, представленный на рисунке 3.1.

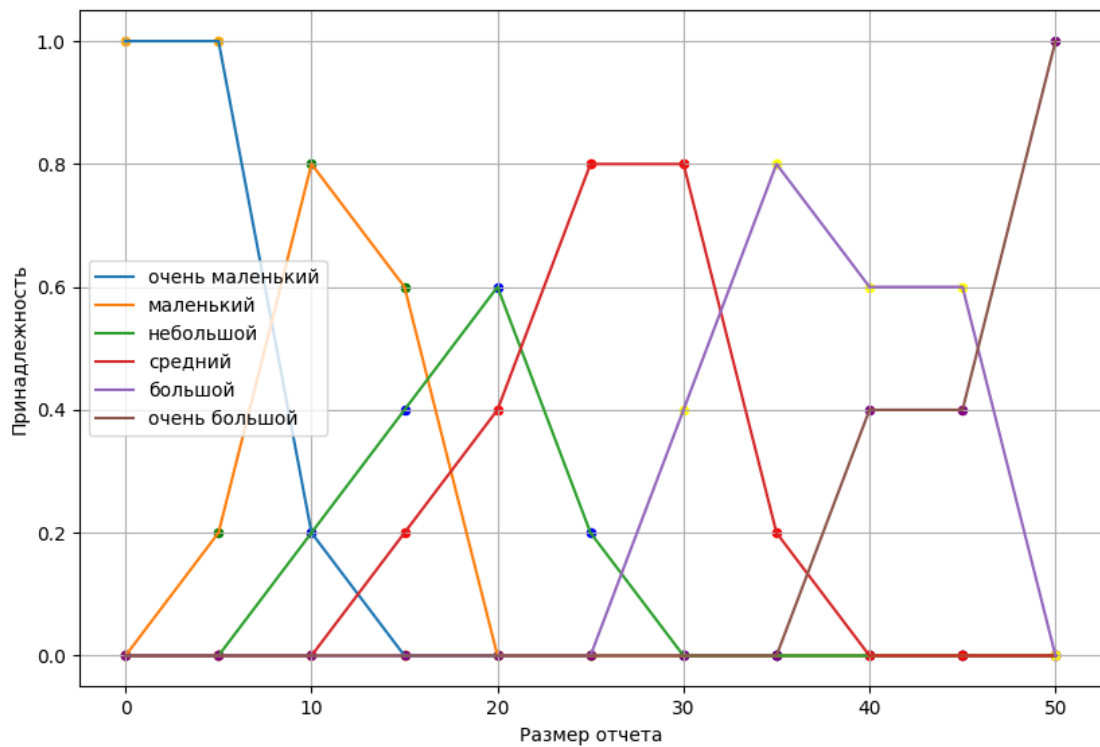


Рисунок 3.1 — Функция принадлежности для лингвистической переменной «размер отчета по Анализу алгоритмов»

Исходя из рисунка 3.1 термы относятся к интервалам как:

- 1) «очень маленький» – от 0 до 10;
- 2) «маленький» – от 10 до 15;
- 3) «небольшой» – от 15 до 20;
- 4) «средний» – от 20 до 35;
- 5) «большой» – от 35 до 45;
- 6) «очень большой» – от 45 до 50.

3.3 Требования к ПО

Программное обеспечение должно удовлетворять следующим требованиям:

- программа принимает на вход запрос к словарю;
- программа выдает массив удовлетворяющих запросу отчетов.

3.4 Средства реализации

В качестве языка программирования для реализации данной лабораторной работы был выбран ЯП Python [5].

Данный язык достаточно удобен и гибок в использовании.

В качестве среды разработки выбор сделан в сторону Visual Studio Code. Данная среда подходит как для Windows, так и для Linux.

3.5 Листинг программы

В приведенном ниже листинге представлена реализация алгоритма полного перебора (листинг 3.1).

Листинг 3.1 — Реализация алгоритма полного перебора

```
1 def find_all(begin, end, not_s = -1, not_e = -1):
2     res = []
3     for l in data.keys():
4         if not_s == -1 and not_e == -1:
5             if data.get(l) >= begin and data.get(l) <= end:
6                 res.append([l, data.get(l)])
7         else:
8             if data.get(l) >= begin and data.get(l) <= not_s:
9                 res.append([l, data.get(l)])
10            elif data.get(l) >= not_e and data.get(l) <= end:
11                res.append([l, data.get(l)])
12    return res
```

В приведенном ниже листинге представлена реализация алгоритма поиска для каждого термина (листинг 3.2).

Листинг 3.2 — Реализация алгоритма поиска по термам

```
1 def find_data(request_raz , request_term):
2
3     answer_arrays = []
4     for c in request_term:
5         if 'не' in c:
6             if 'большой' in c:
7                 answer_arrays.append(find_all(0,50, 35, 45))
8             if 'очень большой' in c:
9                 answer_arrays.append(find_all(0,45))
10            if 'средний' in c:
11                answer_arrays.append(find_all(0,50,25,35))
12            if 'небольшой' in c:
13                answer_arrays.append(find_all(0,50, 15,20))
14            if 'маленький' in c:
15                answer_arrays.append(find_all(0,50, 10,15))
16            if 'очень маленький' in c:
17                answer_arrays.append(find_all(0,50, 0,10))
18        else:
19            if 'большой' in c:
20                answer_arrays.append(find_all(35, 45))
21            if 'очень большой' in c:
22                answer_arrays.append(find_all(45,50))
23            if 'средний' in c:
24                answer_arrays.append(find_all(25,35))
25            if 'небольшой' in c:
26                answer_arrays.append(find_all(15,20))
27            if 'маленький' in c:
28                answer_arrays.append(find_all(10,15))
29            if 'очень маленький' in c:
30                answer_arrays.append(find_all(0,10))
31    answer = []
32    i = 0
33    for raz in request_raz:
34        if 'и' in raz and 'или' not in raz:
35            if i == 0:
36                for a in answer_arrays[i]:
37                    if a in answer_arrays[i+1]:
38                        answer.append(a)
39                i += 2
```

Листинг 3.3 — Реализация алгоритма поиска по термам

```
1         else :
2             for a in answer :
3                 if a not in answer_arrays[i] :
4                     answer.remove(a)
5
6             i+=1
7         if 'или' in raz :
8             if i == 0:
9                 for a in answer_arrays[i] :
10                     answer.append(a)
11                 for a in answer_arrays[i+1]:
12                     if a not in answer :
13                         answer.append(a)
14                 i += 2
15             else :
16                 for a in answer_arrays[i] :
17                     if a not in answer :
18                         answer.append(a)
19                 i+=1
20         if len(request_raz) == 0:
21             for a in answer_arrays[0]:
22                 answer.append(a)
23     return answer
```

3.6 Тестирование ПО

Результаты тестирования ПО приведены на рисунках 3.2, 3.3, 3.4.

```
[macbook@61 lab_07 % python3 main.py
Введите запрос: найти большие и очень большие отчеты
Результаты поиска:
Отчет №14 с размером 45
Отчет №16 с размером 45
Отчет №35 с размером 45
Отчет №47 с размером 45
Отчет №66 с размером 45
Отчет №127 с размером 45
Отчет №149 с размером 45
Отчет №153 с размером 45
Отчет №238 с размером 45
Отчет №335 с размером 45
Отчет №345 с размером 45
Отчет №408 с размером 45
Отчет №413 с размером 45
Отчет №472 с размером 45
Отчет №483 с размером 45
Отчет №570 с размером 45
Отчет №582 с размером 45
Отчет №699 с размером 45
Отчет №737 с размером 45
Отчет №761 с размером 45
Отчет №777 с размером 45
Отчет №795 с размером 45
```

Рисунок 3.2 — Пример запроса пользователя


```

[macbook@61 lab_07 % python3 main.py
Введите запрос: Очень маленький отчет
Результаты поиска:
Отчет №1 с размером 2
Отчет №11 с размером 4
Отчет №21 с размером 5
Отчет №24 с размером 9
Отчет №25 с размером 6
Отчет №37 с размером 3
Отчет №41 с размером 5
Отчет №46 с размером 9
Отчет №50 с размером 4
Отчет №51 с размером 1
Отчет №52 с размером 1
Отчет №55 с размером 10
Отчет №57 с размером 6
Отчет №70 с размером 1
Отчет №76 с размером 5
Отчет №81 с размером 7
Отчет №88 с размером 2
Отчет №92 с размером 2
Отчет №96 с размером 7
Отчет №102 с размером 9
Отчет №103 с размером 10
Отчет №109 с размером 8
Отчет №120 с размером 1
Отчет №121 с размером 4
Отчет №128 с размером 6
Отчет №133 с размером 8
Отчет №142 с размером 6
Отчет №158 с размером 8

```

Рисунок 3.3 — Пример запроса пользователя

```

[macbook@61 lab_07 % python3 main.py
Введите запрос: очень маленькие или очень большие отчеты
Результаты поиска:
Отчет №1 с размером 2
Отчет №11 с размером 4
Отчет №21 с размером 5
Отчет №24 с размером 9
Отчет №25 с размером 6
Отчет №37 с размером 3
Отчет №41 с размером 5
Отчет №46 с размером 9
Отчет №50 с размером 4
Отчет №51 с размером 1
Отчет №52 с размером 1
Отчет №55 с размером 10
Отчет №57 с размером 6
Отчет №70 с размером 1
Отчет №76 с размером 5
Отчет №81 с размером 7
Отчет №88 с размером 2
Отчет №92 с размером 2
Отчет №96 с размером 7
Отчет №102 с размером 9
Отчет №103 с размером 10
Отчет №109 с размером 8
Отчет №120 с размером 1
Отчет №121 с размером 4
Отчет №128 с размером 6
Отчет №133 с размером 8

```

Рисунок 3.4 — Пример запроса пользователя

Вывод

Было написано и протестировано программное обеспечение для решения поставленной задачи.

Заключение

В рамках данной лабораторной работы была достигнута её цель: получен навык поиска по словарю при ограничении на значение признака, заданном при помощи лингвистической переменной. Также выполнены следующие задачи:

- формализован объект и его признак;
- составлена анкета для её заполнения респондентами;
- проведено анкетирование респондентов;
- построена функция принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе статистической обработки мнений респондентов, выступающих в роли экспертов;
- описаны 4 типовых вопросов на русском языке, имеющих целью запрос на поиск в словаре;
- описан алгоритм поиска полным перебором в словаре объектов, удовлетворяющих ограничению, заданному в вопросе на ограниченном естественном языке;
- описана структуру данных словаря, хранящего наименования объектов согласно варианту и числовое значение признака объекта;
- реализован алгоритм поиска полным перебором в словаре;
- приведены примеры запросов пользователя и сформированный реализацией алгоритм поиска выборки объектов из словаря, используя составленные респондентами вопросы.

Предложенный в лабораторной работе алгоритм ограничено применим только в рамках поставленной задачи. Имеется возможность корректировать термы под определенную задачу, но только для универсального множества в виде целых чисел. Ограничения данного алгоритма заключаются в том, что он не приспособлен для запросов не включающих установленных термов.

Список источников

- 1) Оразбаев Б. Б., Курмангазиева Л. Т., Коданова Ш. К. “ТЕОРИЯ И МЕТОДЫ СИСТЕМНОГО АНАЛИЗА”, 2017.
- 2) National Institute of Standards and Technology [Электронный ресурс].
Режим доступа: <https://xlinux.nist.gov/dads/HTML/assocarray.html> (дата обращения 17.12.2022).
- 3) Н. Нильсон. Искусственный интеллект. Методы поиска решений. М.: Мир, 1973. с. 273.
- 4) dict Python [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/library/2to3.html?highlight=dict#to3fixer-dict> (дата обращения: 17.12.2022).
- 5) Welcome to Python [Электронный ресурс]. Режим доступа: <https://www.python.org> (дата обращения: 17.12.2022).
- 6) list Python [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/library/pdb.html?highlight=list#pdbcommand-list> (дата обращения: 17.12.2022).