



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 по дисциплине «Анализ алгоритмов»

Тема: Поиск в словаре

Студент: Карпова Е. О.

Группа: ИУ7-52Б

Оценка (баллы): \_\_\_\_\_

Преподаватели: Волкова Л. Л., Строганов Ю. В.

Москва — 2022 г.

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Аналитическая часть</b>	<b>5</b>
1.1. Формализация объекта и выбранного признака классификации . . . . .	5
1.1.1. Применяемые правила нечёткой логики . . . . .	5
1.1.2. Алгоритм бинарного поиска в словаре . . . . .	6
<b>2. Конструкторская часть</b>	<b>7</b>
2.1. Разработка алгоритма бинарного поиска в словаре . . . . .	7
2.2. Разработка алгоритма определения соответствия введённого запроса распо- знаваемым вариантам . . . . .	8
2.3. Разработка структуры данных словаря . . . . .	9
<b>3. Технологическая часть</b>	<b>10</b>
3.1. Требования к ПО . . . . .	10
3.2. Требования к входному запросу . . . . .	10
3.3. Средства реализации . . . . .	11
3.4. Реализация алгоритмов . . . . .	11
3.5. Реализация структуры словаря . . . . .	14
3.6. Тестирование . . . . .	15
3.7. Интерфейс веб-сервиса . . . . .	16
<b>4. Исследовательская часть</b>	<b>17</b>
4.1. Анкета для респондентов . . . . .	17
4.2. Результаты анкетирования . . . . .	17
4.3. Функция принадлежности термам числовых значений признака, описывае- мого лингвистической переменной . . . . .	19
4.4. Соответствие признаков и диапазонов значений . . . . .	21
<b>Заключение</b>	<b>22</b>
<b>Список использованных источников</b>	<b>23</b>

# Введение

В современном мире постоянно растёт объём доступной информации. С информацией необходимо взаимодействовать — получать, анализировать, хранить, осуществлять её поиск, транслировать.

Если определённые блоки информации можно идентифицировать по определённому единичному значению, то подходящим методом хранения информации будет словарь. Словарь, или ассоциативный массив, означает абстрактный тип данных формата «ключ-значение». То есть одному заданному признаку ставятся в соответствие данные, определённым образом соотносящиеся с данным признаком.

Такой вариант хранения данных пригоден, если требуется осуществлять их периодический поиск, поэтому на основе словаря можно сформировать своеобразную поисковую систему. Однако, пользовательский ввод зачастую неточен, предлагаемые категории признака абстрактны и требуют конкретизации. Подобная экспертная информация трудно формализуема в рамках стандартных математических подходов. В связи с этим, необходимо воспользоваться правилами нечёткой логики [8].

Цель работы: получение навыков программирования, тестирования полученного программного продукта и поиска по словарю при ограничении на значение признака, заданного при помощи лингвистической переменной.

Задачи работы:

- 1) выбор объекта и признака объекта для разделения на категории;
- 2) формализация выбранного признака и определение категорий;
- 3) составление анкеты для респондентов;
- 4) проведение опроса среди респондентов по составленной анкете;
- 5) построение функции принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе проведения статистического анализа мнений респондентов;
- 6) описание допустимых формулировок при формировании запроса;

- 7) описание алгоритма поиска в словаре сущностей, удовлетворяющих ограничению, заданному в запросе на ограниченном естественном языке, и алгоритма определения соответствия введённого запроса распознаваемым вариантам;
- 8) описание структуры данных словаря;
- 9) реализация выбранного алгоритма поиска в словаре;
- 10) анализ применимости предложенного алгоритма.

# 1. Аналитическая часть

В данном разделе будут рассмотрены применяемые правила нечёткой логики, представлена формализация объекта и выбранного признака классификации и описан алгоритм бинарного поиска в словаре.

## 1.1. Формализация объекта и выбранного признака классификации

Словарь — абстрактный тип данных формата «ключ-значение». То есть одному заданному признаку ставятся в соответствие данные, определённым образом соотносящиеся с данным признаком.

В данной работе ключом словаря является количество шерсти у кошки на  $\text{см}^2$  тела. Иными словами, степень пушистости кошки. Значением является структура, хранящая информацию о породе кошки, её пушистости и ссылке на изображение кошки. При поиске значения в словаре осуществляется поиск по ключу. Данные в словаре хранятся в отсортированном порядке.

### 1.1.1. Применяемые правила нечёткой логики

Представление экспертных знаний в конкретной предметной области предполагает определение конечного множества лингвистических переменных (в данной работе — одна, пушистость кошки), термов для лингвистической переменной, построение функции принадлежности и определение диапазонов значений признака, соответствующих каждому терму.

Задаётся множество термов

$$T'_i = (t_1, t_2, \dots, t_m), \quad (1.1)$$

где  $m$  — количество термов.

Также задаётся множество значений величины, которую характеризует лингвистическая переменная:

$$X = (x_1, x_2, \dots, x_n) \quad (1.2)$$

где  $n$  — количество значений.

Для каждого  $i = \overline{1, m}, j = \overline{1, n}$  требуется определить значение  $\mu_{t_i}(x_j)$  — степени функции принадлежности элементов множества  $X$  к элементам из множества  $T'_i$ .

В данной работе данное значение определяется при помощи анализа статистической выборки по результатам опроса респондентов (экспертов).

Тогда степень функции принадлежности элементов множества  $X$  к элементам из множества  $T'_i$  можно рассчитать по формуле

$$\mu_{t_i}(x_j) = \frac{1}{k} \cdot \sum_{k=1}^K a_{ji}^k, \quad (1.3)$$

где  $K$  — количество экспертов,  $a_{nm}^k$  — бинарная экспертная оценка  $k$ -м экспертом у элемента  $x_j$  свойств нечёткого множества  $T'_i$ ,  $i = \overline{1, m}$ .

По полученным значениям для каждого терма строится функция принадлежности, для которой по оси абсцисс откладывается значение величин из  $X$ , а по оси ординат — значение степени функции принадлежности элементов множества  $X$  к элементам из множества  $T'_i$ .

Терм будет определять те значения из диапазона, для которых график его функции принадлежности будет расположен выше остальных.

### 1.1.2. Алгоритм бинарного поиска в словаре

Бинарный поиск широко используется в информатике, вычислительной математике и программировании [9]. Суть данного алгоритма — нахождение в массиве (в условиях данной лабораторной работы — в словаре) элемента по ключу. Изначальная отсортированность данных — важное условие работы алгоритма.

В данной работе бинарный поиск используется последовательно два раза — для определения начального и конечного значений для диапазона элементов из множества  $X$  для терма.

В данном алгоритме применяется последовательное деление массива и его частей на половины на каждой итерации цикла, то есть, постепенное приближение к искомому значению. Значение считается найденным при схождении границ обзора массива, которые в начале работы алгоритма устанавливаются равными 0 и индексу последнего элемента в наборе соответственно. Вторым случаем является завершение работы алгоритма при пересечении границ обзора, что случается, когда в наборе нет искомого элемента. В контексте данной работы такая ситуация также валидна. Тогда в качестве результата выбирается большее из значений границ, если ведётся поиск начального значения для диапазона элементов множества  $X$ , и меньшее, если конечного.

## 2. Конструкторская часть

В этом разделе разработаны структура данных словаря, алгоритм бинарного поиска в словаре и алгоритм определения соответствия введённого запроса распознаваемым вариантам. Также будут представлены схемы данных алгоритмов.

### 2.1. Разработка алгоритма бинарного поиска в словаре

На рисунке 2.1 представлена схема реализации алгоритма бинарного поиска в словаре.

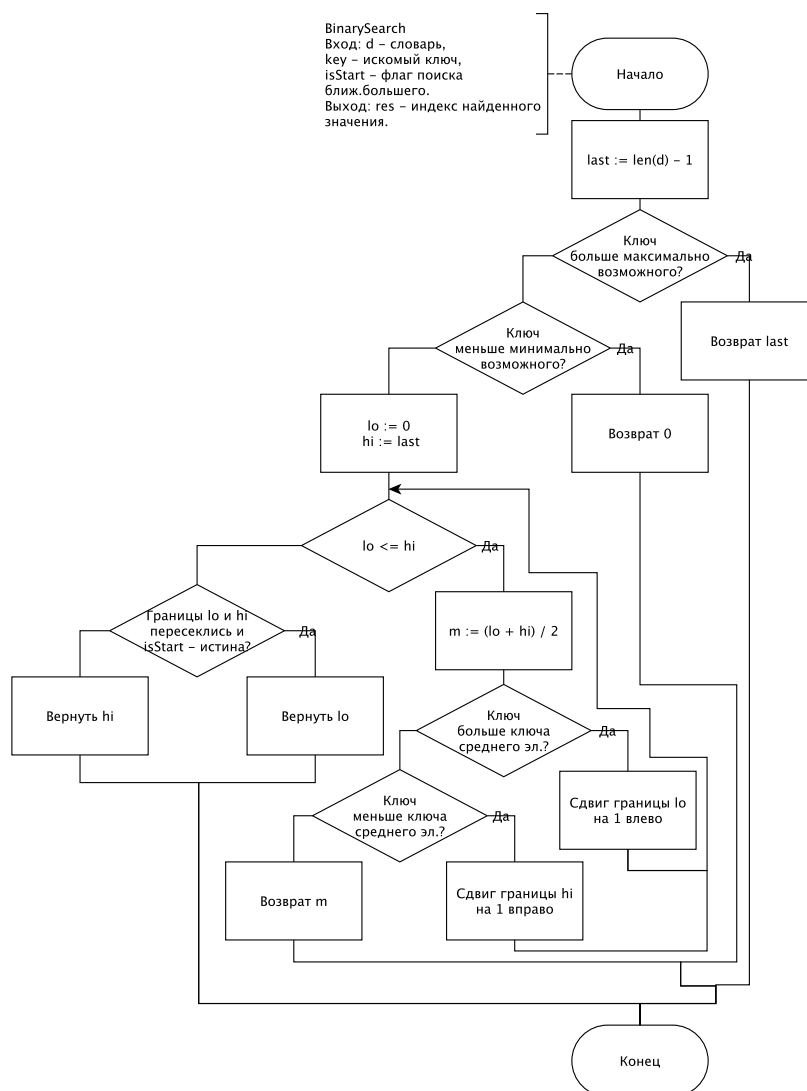


Рисунок 2.1 — Схема реализации алгоритма бинарного поиска в словаре

## 2.2. Разработка алгоритма определения соответствия введённого запроса распознаваемым вариантам

На рисунке 2.2 представлена схема реализации алгоритма определения соответствия введённого запроса распознаваемым вариантам.

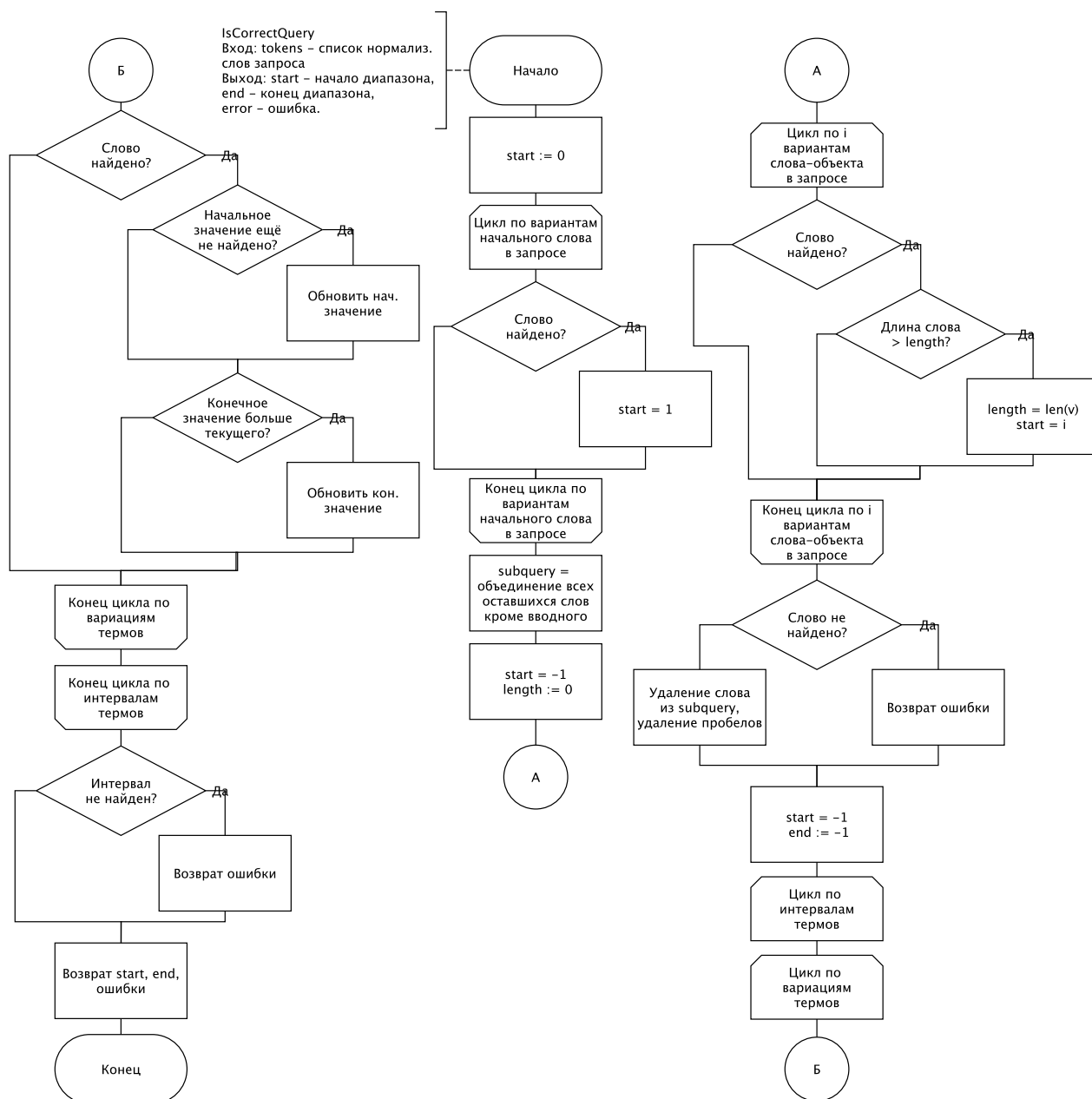


Рисунок 2.2 — Схема реализации алгоритма определения соответствия введённого запроса распознаваемым вариантам



## 2.3. Разработка структуры данных словаря

Было принято решение подразумевать под структурой словаря массив *Dictionary* записей *Word* формата «ключ-значение», где ключом *key* словаря является количество шерсти у кошки на см<sup>2</sup> тела, а значением *value* является структура, хранящая информацию о породе кошки, её пушистости и ссылку на изображение кошки.

## 3. Технологическая часть

В данном разделе будут представлены реализации алгоритма бинарного поиска в словаре и алгоритма определения соответствия введённого запроса распознаваемым вариантам. Также будут указаны обязательные требования к ПО, средства реализации алгоритмов, представлены интерфейс разработанного веб-сервиса и результаты проведённого тестирования программы.

### 3.1. Требования к ПО

Для программы выделен перечень требований:

- программой предоставляется интерфейс в формате веб-сервиса с поисковой строкой;
- программой обрабатываются запросы, сформированный по указанным ниже требованиям;
- программой не воспринимаются запросы, не соответствующие требованиям, выводится сообщение об ошибке;
- программой производится аварийное завершение с текстом об ошибке при иных ошибках;
- программой выводится список удовлетворяющих запросу данных с иллюстрациями при валидном запросе.

### 3.2. Требования к входному запросу

Запрос может быть воспринят программой, как валидный, если он удовлетворяет следующим правилам:

- «выведи», «дай», «показывай», «покажи», «какие», «выдай», «хочу», «найти», «найди» в качестве вводного слова, при этом само наличие вводного слова является обязательным в запросе;
- «кошка», «кот», «котёнок», «кошечка», «котик», «котёночек» и другие формы этих слов, приводимые при нормализации к указанной, при этом наличие подобного указания на тип объекта является обязательным;

- Указанные далее категории в начальной или иной форме слова; категории объединены в классы эквивалентности символом «-»: «лысый»-«не пушистый», «плешивый»-«не особо пушистый»-«не пушистый»-«не лысый», «волосатый»-«не особо пушистый»-«не пушистый»-«не очень пушистый»-«не лысый», «пушистый»-«не лысый»-«немного пушистый», «очень пушистый»-«не лысый»-«сильно пушистый», «невероятно пушистый»-«не лысый»;
- вводное слово обязано быть первым в запросе, если оно есть, порядок слова-объекта и слова-категории относительно друг друга может быть любым.

### 3.3. Средства реализации

Для реализации данной работы был выбран язык программирования Go [1]. Выбор обусловлен наличием в *Go* библиотек для тестирования ПО, работы с *http* протоколом [7], а также необходимых для реализации поставленных цели и задач средств. В качестве среды разработки была выбрана *GoLand* [3]. Для хранения данных о котах была создана база данных *cats* с помощью СУБД *Postgres* [10].

### 3.4. Реализация алгоритмов

В листингах 3.1 – 3.5 представлены реализации алгоритма бинарного поиска в словаре и алгоритма определения соответствия введённого запроса распознаваемым вариантам.

Листинг 3.1 — Листинг функции реализации алгоритма бинарного поиска в словаре  
(начало)

```
func (d *Dictionary) BinarySearch(key int, isStart bool) int {
    last := len(*d) - 1
    if key > (*d)[last].Key {
        return last
    }
    if key < (*d)[0].Key {
        return 0
    }
    lo := 0
    hi := last
```

Листинг 3.2 — Листинг функции реализации алгоритма бинарного поиска в словаре  
(продолжение листинга 3.1)

```
    for lo <= hi {
        m := (lo + hi) / 2
        if key > (*d)[m].Key {
            lo = m + 1
        } else if key < (*d)[m].Key {
            hi = m - 1
        } else {
            return m
        }
    }
    crossed := (*d)[lo].Key > (*d)[hi].Key
    if crossed == isStart {
        return lo
    }
    return hi
}
```

Листинг 3.3 — Листинг функции алгоритма определения соответствия введённого  
запроса распознаваемым вариантам (начало)

```
func IsCorrectQuery(tokens []string) (int, int, error) {
    start := 0

    for _, v := range Starters {
        if tokens[0] == v {
            start = 1
            break
        }
    }

    subquery := strings.Join(tokens[start:], " ")

    start = -1
    length := 0
}
```

Листинг 3.4 — Листинг функции алгоритма определения соответствия введённого запроса распознаваемым вариантам (продолжение листинга 3.3)

```
for i, v := range Objects {
    if strings.Contains(subquery, v) {
        if len(v) > length {
            length = len(v)
            start = i
        }
    }
}

if start == -1 {
    return -1, -1, errors.New("неверный объект запроса")
} else {
    subquery = strings.Replace(subquery, Objects[start], "", 1)
    subquery = strings.TrimSpace(subquery)
}

start = -1
end := -1

for _, v := range Intervals {
    for _, e := range v.Names {
        if subquery == e {
            if start == -1 {
                start = v.Start
            }
            if v.End > end {
                end = v.End
            }
        }
    }
}
}
```

Листинг 3.5 — Листинг функции алгоритма определения соответствия введённого запроса распознаваемым вариантам (продолжение листинга 3.4)

```
    if start == -1 || end == -1 {  
        return -1, -1, errors.New("некорректный формат запроса")  
    }  
  
    return start, end, nil  
}
```

### 3.5. Реализация структуры словаря

В листинге 3.6 представлена реализации структуры словаря.

Листинг 3.6 — Листинг реализации структуры словаря

```
type Word struct {  
    Key    int  
    Value models.Cat  
}  
  
type Dictionary []Word
```

## 3.6. Тестирование

В таблице 3.1 представлены тесты запросов для поиска в словаре. Все тесты пройдены успешно.

Таблица 3.1 — Тесты для поиска в словаре

№	Запрос	Результат
1	найти плешивых котиков	Ориентальная Количество пушинок на квадратный см – 424! ...
2	дай пушистых котят	Нибелунг Количество пушинок на квадратный см – 8777! ...
3	невероятно пушистые кошечки	Сибирская Количество пушинок на квадратный см – 18005! ...
4	невероятно пушистые рыбки	Ошибка выполнения запроса: неверный объект запроса
5	покажи короче котов мне быстро	Ошибка выполнения запроса: некорректный формат запроса

## 3.7. Интерфейс веб-сервиса

На рисунках 3.1 – 3.2 представлен интерфейс разработанного веб-сврвиса.

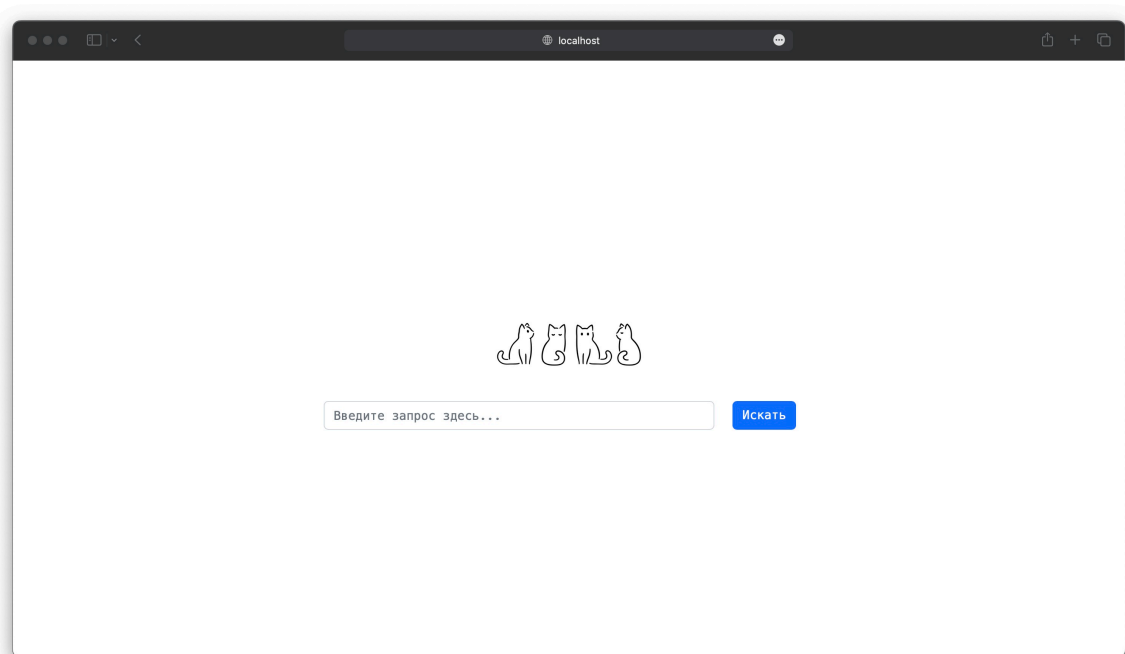


Рисунок 3.1 — Интерфейс разработанного веб-сврвиса: домашняя страница

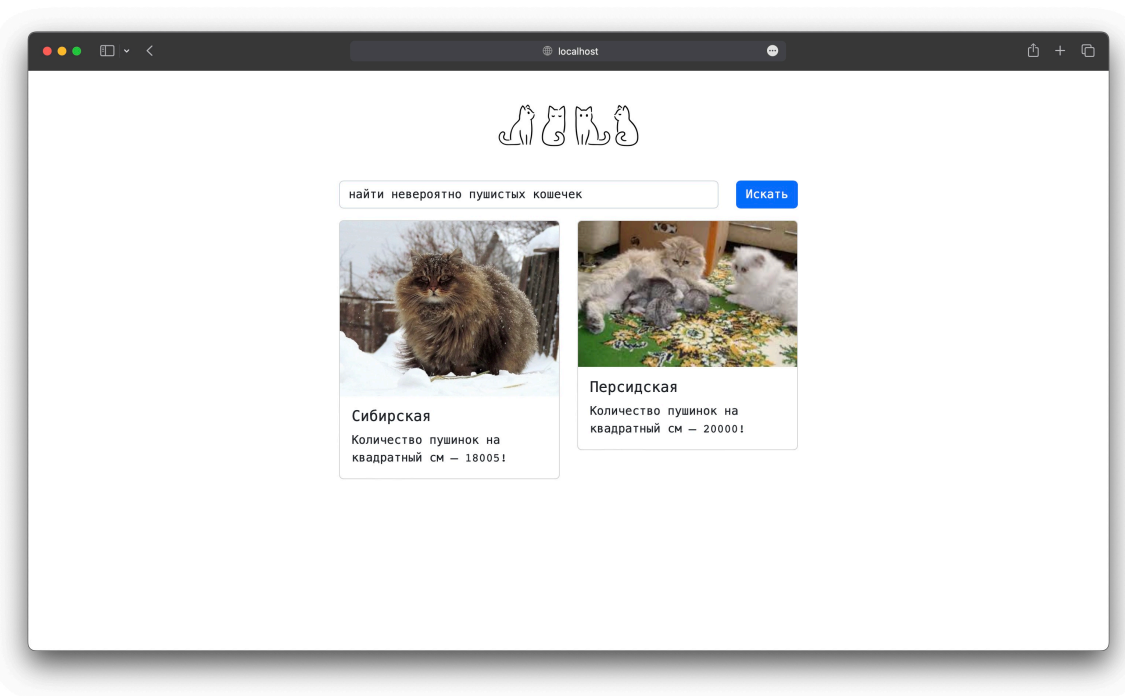


Рисунок 3.2 — Интерфейс разработанного веб-сврвиса: страница поиска



## 4. Исследовательская часть

Для формирования системы запросов о пушистости кошек, было необходимо провести опрос среди респондентов и построить функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной — пушистостью кошки.

В данном разделе приведена анкета, предоставленная респондентам. Также представлены результаты анкетирования и статистической обработки мнений респондентов.

### 4.1. Анкета для респондентов

Анкета, отправленная респондентам, представляет собой таблицу.

Таблица 4.1 — Анкета, отправленная респондентам

№ респондента	Терм $i$	Количество пушинок на $\text{см}^2$ $\xi$ , шт.						
		0	500	1000	5000	10000	15000	20000
1	лысая							
	плешивая							
	волосатая							
	пушистая							
	очень пушистая							
	невероятно пушистая							

### 4.2. Результаты анкетирования

В таблице 4.2 приведено соответствие между респондентами и их номером.

Таблица 4.2 — Соответствие № респондента и респондента

№	Респондент
1	Княжев А. В.
2	Глотов И. А.
3	Ляпина Н. В.
4	Аскарян К. А.
5	Обревская В. В.

В таблице 4.3 приведены результаты анкетирования респондентов.

Таблица 4.3 — Результаты анкетирования респондентов

№ респондента	Терм $i$	Количество пушинок на $\text{см}^2$ $\xi$ , шт.						
		0	500	1000	5000	10000	15000	20000
1	лысая	1	0	0	0	0	0	0
	плешивая	0	1	0	0	0	0	0
	волосатая	0	0	0	0	0	0	0
	пушистая	0	0	1	1	0	0	0
	очень пушистая	0	0	0	0	1	1	0
	невероятно пушистая	0	0	0	0	0	0	1
2	лысая	1	1	0	0	0	0	0
	плешивая	0	0	1	0	0	0	0
	волосатая	0	0	0	1	0	0	0
	пушистая	0	0	0	0	1	0	0
	очень пушистая	0	0	0	0	0	1	0
	невероятно пушистая	0	0	0	0	0	0	1
3	лысая	1	0	0	0	0	0	0
	плешивая	0	1	1	0	0	0	0
	волосатая	0	0	0	1	0	0	0
	пушистая	0	0	0	0	1	0	0
	очень пушистая	0	0	0	0	0	1	0
	невероятно пушистая	0	0	0	0	0	0	1
4	лысая	1	1	0	0	0	0	0
	плешивая	0	0	0	0	0	0	0
	волосатая	0	0	1	1	1	0	0
	пушистая	0	0	0	0	0	1	0
	очень пушистая	0	0	0	0	0	0	1
	невероятно пушистая	0	0	0	0	0	0	0
5	лысая	1	0	0	0	0	0	0
	плешивая	0	1	0	0	0	0	0
	волосатая	0	0	1	0	0	0	0
	пушистая	0	0	0	1	0	0	0
	очень пушистая	0	0	0	0	0	0	0
	невероятно пушистая	0	0	0	0	1	1	1

### 4.3. Функция принадлежности термам числовых значений признака, описываемого лингвистической переменной

Из результатов анкетирования респондентов получим функцию принадлежности термам числовых значений признака, описываемого лингвистической переменной.

Таблица 4.4 — Таблица значений функции принадлежности

лысая	$\mu_j(\xi)$	5	2	0	0	0	0	0
	$F_j(\xi)$	1.0	0.4	0.0	0.0	0.0	0.0	0.0
плешивая	$\mu_j(\xi)$	0	3	2	0	0	0	0
	$F_j(\xi)$	0.0	0.6	0.4	0.0	0.0	0.0	0.0
волосатая	$\mu_j(\xi)$	0	0	2	3	1	0	0
	$F_j(\xi)$	0.0	0.0	0.4	0.6	0.2	0.0	0.0
пушистая	$\mu_j(\xi)$	0	0	1	2	2	1	1
	$F_j(\xi)$	0.0	0.0	0.2	0.4	0.4	0.2	0.2
очень пушистая	$\mu_j(\xi)$	0	0	0	0	1	3	1
	$F_j(\xi)$	0.0	0.0	0.0	0.0	0.2	0.6	0.2
невероятно пушистая	$\mu_j(\xi)$	0	0	0	0	1	1	3
	$F_j(\xi)$	0.0	0.0	0.0	0.0	0.2	0.2	0.6

В таблице выше  $\mu_j(\xi) = \sum_{k=1}^K a_{ji}^k$ ,  $F_j(\xi) = \frac{S_j(\xi)}{K}$ , где  $K$  — количество респондентов.

На рисунках 4.1 и 4.2 представлены графики, отражающий функциональную зависимость  $F_j(\xi)$ .

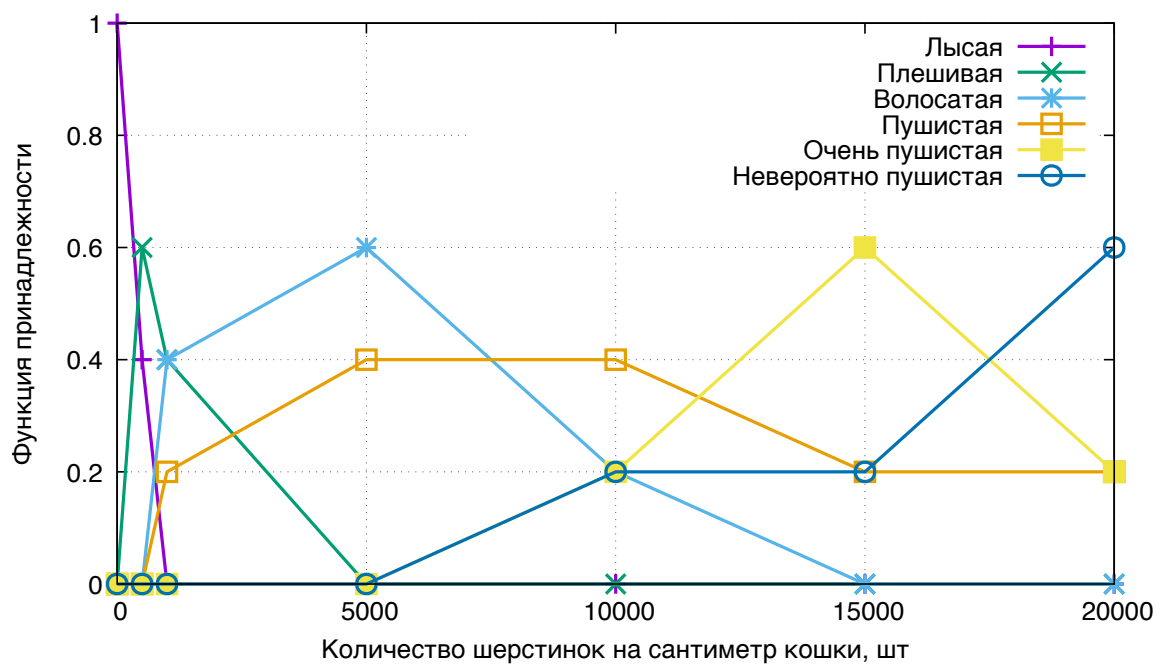


Рисунок 4.1 — Функциональная зависимость  $F_j(\xi)$  (от 0 до 5000 по оси абсцисс)

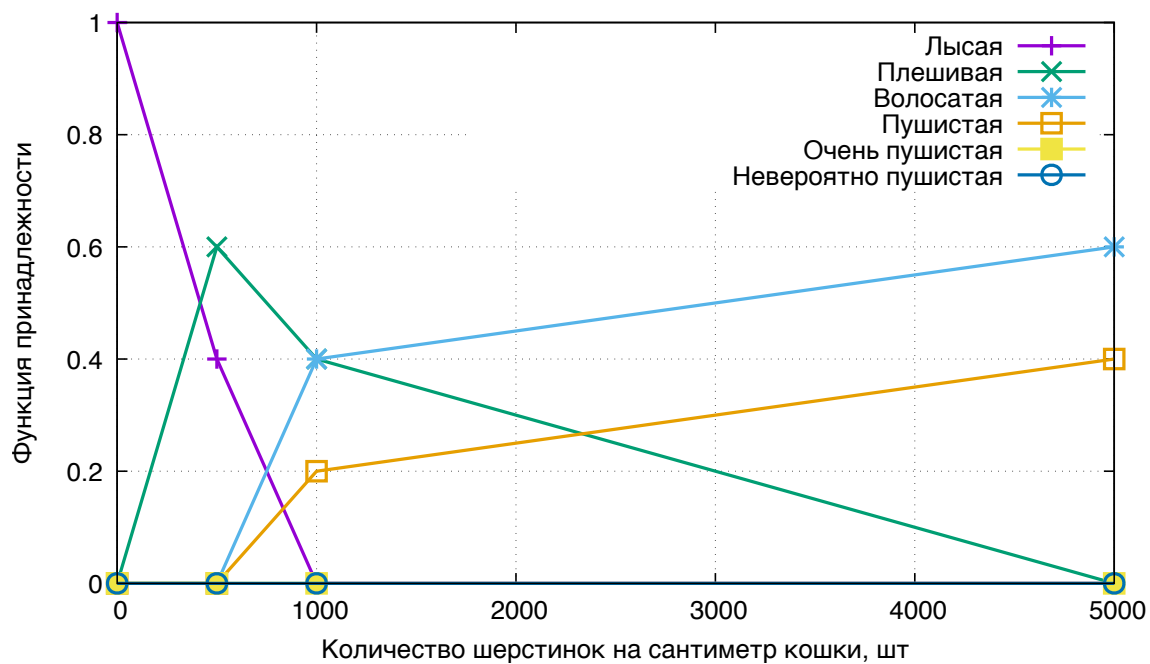


Рисунок 4.2 — Функциональная зависимость  $F_j(\xi)$

## 4.4. Соответствие признаков и диапазонов значений

В таблице 4.5 приведено соответствие между признаками и диапазонами значений пушистости кошки.

Таблица 4.5 — Соответствие между терминами и диапазонами значений пушистости кошки

Признак	Диапазон
лысая	[0; 417]
плешивая	[418; 1000]
волосатая	[1001; 7500]
пушистая	[7501; 11667]
очень пушистая	[11668; 17500]
невероятно пушистая	[17500; 20000]

# Заключение

Алгоритм бинарного поиска применительно к задаче поиска элемента в словаре по ключу достаточно эффективен, так как имеет временную сложность  $O(\log 2n)$  [11]. Однако, ограничением эффективности данного алгоритма является эффективность предварительно примененной сортировки: если выбранный алгоритм сортировки неэффективен, то в целом поиск по словарю также займёт большое количество времени, что будет неэффективно.

В ходе выполнения лабораторной работы была достигнута поставленная цель: были получены навыки программирования, тестирования полученного программного продукта и поиска по словарю при ограничении на значение признака, заданного при помощи лингвистической переменной.

В процессе выполнения лабораторной работы были также реализованы все поставленные задачи, а именно:

- был произведён выбор объекта и признака объекта для разделения на категории;
- была выполнена формализация выбранного признака и определение категорий;
- была составлена анкета для респондентов;
- был проведён опрос среди респондентов по составленной анкете;
- была построена функция принадлежности термам числовых значений признака, описываемого лингвистической переменной, на основе проведения статистического анализа мнений респондентов;
- были описаны допустимые формулировки при формировании запроса;
- был описан алгоритм поиска в словаре сущностей, удовлетворяющих ограничению, заданному в запросе на ограниченном естественном языке, и алгоритм определения соответствия введённого запроса распознаваемым вариантам;
- была описана структура данных словаря;
- были реализованы описанные алгоритмы;
- был проведён анализ применимости предложенного алгоритма.

# Список использованных источников

1. Документация по языку программирования *Go* [Электронный ресурс]. Режим доступа: <https://go.dev/doc/> (дата обращения: 20.09.2022).
2. Документация по пакетам языка программирования *Go* [Электронный ресурс]. Режим доступа: <https://pkg.go.dev> (дата обращения: 20.09.2022).
3. GoLand: IDE для профессиональной разработки на *Go* [Электронный ресурс]. Режим доступа: <https://www.jetbrains.com/ru-ru/go/> (дата обращения: 20.09.2022).
4. Техническая спецификация ноутбука *MacBook Air* [Электронный ресурс]. Режим доступа: <https://support.apple.com/kb/SP869> (дата обращения: 20.09.2022).
5. *Apple M2* [Электронный ресурс]. Режим доступа: <https://www.notebookcheck.net/Apple-M2-Processor-Benchmarks-and-Specs.632312.0.html> (дата обращения: 10.10.2022).
6. *gnuplot homepage* [Электронный ресурс]. Режим доступа: <http://www.gnuplot.info> (дата обращения: 10.10.2022).
7. Документация пакета *snet/http* языка *Go* [Электронный ресурс]. Режим доступа: <https://pkg.go.dev/net/http> (дата обращения: 10.10.2022).
8. Борисова Л. В., Нурутдинова И. Н., Димитров В. П. О методике представления нечётких экспертных знаний // *Advanced Engineering Research*. – 2014. – Т. 14. – №. 4 (79). – С. 93-102.
9. Ткачева, Е. А. Алгоритм бинарного поиска / Е. А. Ткачева // *Электронное обучение в непрерывном образовании*. – 2014. – № 1-2. – С. 341-343. – EDN TLSDSP.
10. *PostgreSQL homepage* [Электронный ресурс]. Режим доступа: <https://www.postgresql.org> (дата обращения: 10.10.2022).
11. Абдуллаева Д. Н., Чернова С. В. Обзор алгоритмов линейного и бинарного поиска // *Молодежный научный форум: технические и математические науки*. – 2017. – №. 8. – С. 4-7.