



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

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ
по лабораторной работе № 1
по курсу «Основы искусственного интеллекта»
на тему: «Опрос эксперта»

Студент ИУ7-13М
(Группа)

(Подпись, дата)

Орду М. А.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Строганов Ю. В.
(И. О. Фамилия)

2025 г.

СОДЕРЖАНИЕ

1	Теоретическая часть	3
1.1	Введение	3
1.2	Описание предметной области	3
2	Практическая часть	4
2.1	Особенности программной реализации	4
2.2	Модель базы знаний	4
2.3	Типовой сценарий взаимодействия	4
2.4	ER-диаграмма	7
	ПРИЛОЖЕНИЕ А Исходный код программы	9
	ПРИЛОЖЕНИЕ Б Сформированная база данных	13

1 Теоретическая часть

1.1 Введение

Цель лабораторной работы: создание программы для работы с базой знаний о вкусах и характеристиках различных сортов пива. Задачи работы:

- Провести опрос эксперта о вкусах пива и собрать базу знаний.
- Реализовать базу знаний с различными характеристиками сортов пива.
- Создать приложение для взаимодействия с базой знаний.
- Реализовать не менее трёх запросов к системе с параметрами.

1.2 Описание предметной области

Предметная область включает информацию о сортах пива и их характеристиках. На рисунке 1.1 представлена схема рассматриваемой предметной области.

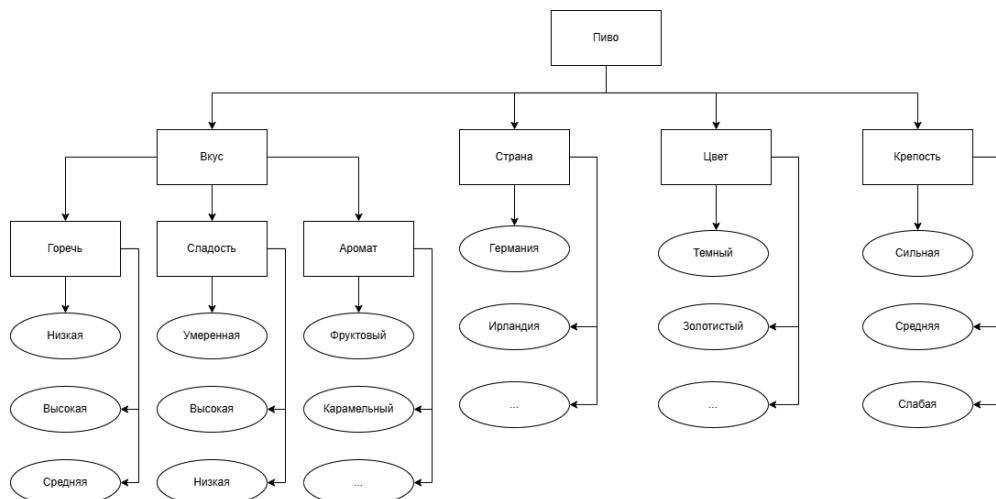


Рисунок 1.1 – Схема предметной области

На основе этих знаний система может рекомендовать сорта пива по выбранным пользователем параметрам.

2 Практическая часть

2.1 Особенности программной реализации

Программная реализация экспертной системы выполнена на языке Python и представляет собой интерактивное консольное приложение. Основой системы служит база знаний `base`, содержащая характеристики различных сортов пива: тип горечи, аромат, цвет, крепость и страну происхождения (замечу, что при опросе эксперта данный пункт фигурировал как 'страна с которой ассоциируется этот сорт пива', поэтому корректность именования этого атрибута 'страной происхождения' подлежит сомнению).

Логика взаимодействия с пользователем реализована через меню, предоставляющее четыре основных режима работы: вывод характеристик выбранного сорта, рекомендация пива по вкусовым параметрам, поиск по стране происхождения и поиск по сочетанию крепости и цвета. Каждый пункт меню связан с отдельной функцией, которая осуществляет обработку пользовательского ввода, фильтрацию данных из базы знаний и вывод результата.

2.2 Модель базы знаний

База знаний реализована с использованием фреймовой модели: каждый сорт пива представлен как фрейм с набором атрибутов.

Листинг 2.1 – Пример объекта в базе данных

```
1 "Лагер": {
2     "горечь": "низкая",
3     "сладость": "низкая",
4     "аромат": "хмелевой",
5     "цвет": "светлый",
6     "крепость": "слабая",
7     "страна": "Германия или Бельгия"
8 }
```

2.3 Типовой сценарий взаимодействия

Пользователь выбирает один из пунктов меню:

1. Показать характеристики пива

2. Рекомендовать пиво по вкусу
3. Найти пиво по стране происхождения
4. Найти пиво по крепости и цвету

На рисунке 2.1 представлена Use case диаграмма программы. В листинге 2.2 представлен пример работы с программой.



Рисунок 2.1 – Use Case диаграмма

Листинг 2.2 – Пример работы с консольной программой

```

1 === Экспертная система: Вкусы пива ===
2 1. Показать характеристики пива
3 2. Рекомендовать пиво по вкусу
4 3. Найти по стране происхождения
5 4. Найти по крепости и цвету
6 0. Выход
7 Выберите действие: 2
8
9 Выберите желаемую горечь:
10 1. низкая
11 2. средняя
12 3. высокая
13 Номер горечи: 2
14
15 Выберите желаемый аромат:
16 1. фруктовый
17 2. цветочный
18 3. шоколадный
19 ...
20 Номер аромата: 1
21
22 Рекомендуется: Эль , Американский Эль

```

2.4 ER-диаграмма

Ниже на рисунке 2.2 представлена диаграмма сущностей и связей в нотации Чена:

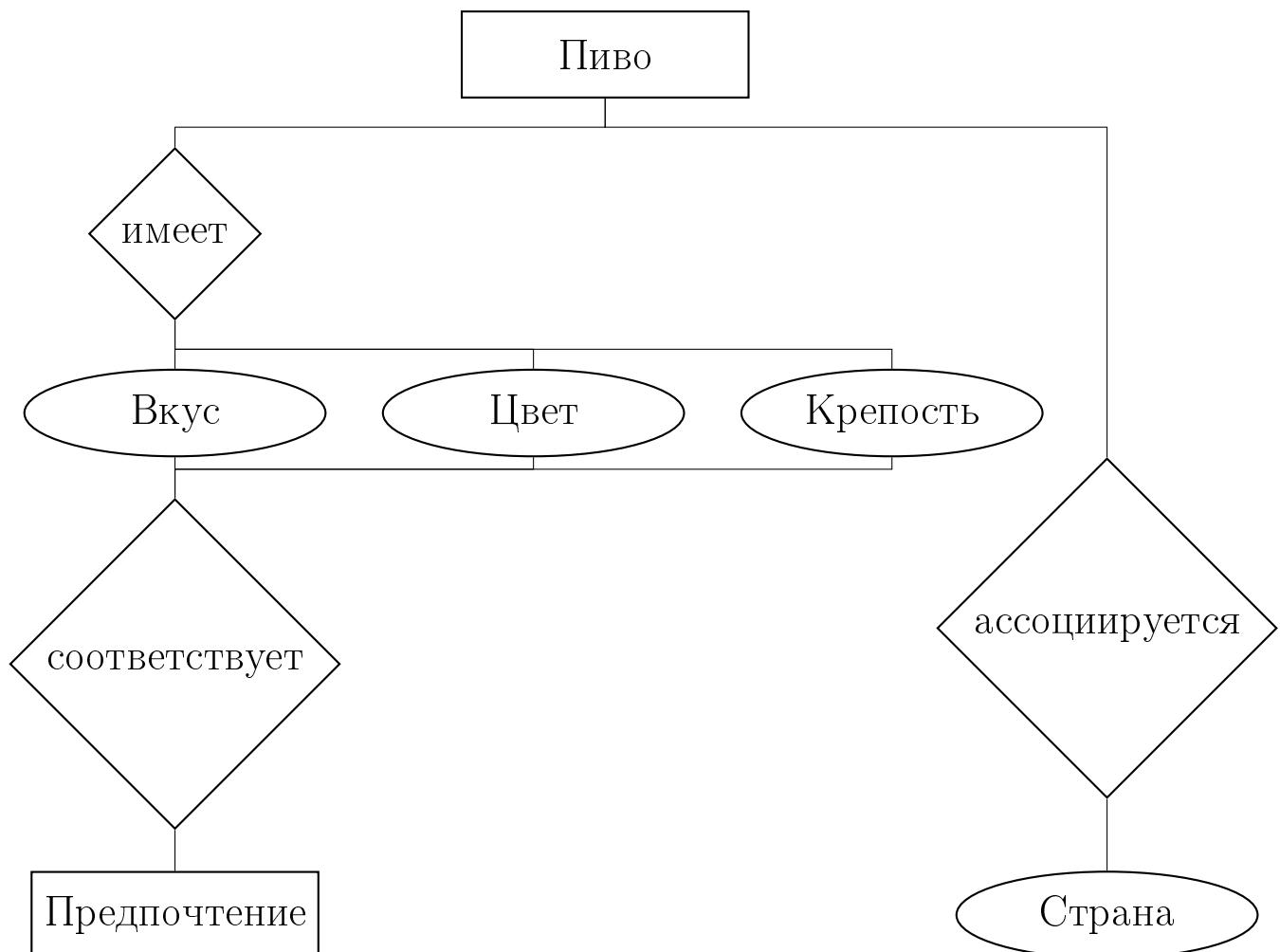


Рисунок 2.2 – ER-диаграмма предметной области

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была разработана экспертная система по сортам пива с фреймовой базой знаний. Система позволяет:

- хранить информацию сортах пива с различными характеристиками;
- делать рекомендации по вкусовым предпочтениям, стране происхождения и крепости/цвету;
- демонстрировать атрибуты конкретного сорта пива пользователю.

ПРИЛОЖЕНИЕ А

Исходный код программы

Листинг А.1 – Исходный код программы

```
1 from base import beer_knowledge_base
2
3 def find_beer_by_name(name):
4     """Вывод характеристик пива по названию."""
5     beer = beer_knowledge_base.get(name)
6     if beer:
7         print(f"\nТип пива: {name}")
8         for k, v in beer.items():
9             print(f"  {k.capitalize()}: {v}")
10    else:
11        print("Такого пива нет в базе.")
12
13 def recommend_beer():
14     """Рекомендация пива по выбранной горечи и аромату."""
15     bitterness_options = ["низкая", "средняя", "высокая"]
16     aroma_options = ["фруктовый", "цветочный", "чистый",
17                      "нейтральный",
18                      "жёлтый", "кофейный", "карамельный",
19                      "цитрусовый", "хвойный", "хмелевой",
20                      "банановый", "гвоздичный"]
21
22     print("\nВыберите желаемую горечь:")
23     for i, b in enumerate(bitterness_options, 1):
24         print(f"{i}. {b}")
25     b_choice = int(input("Номер горечи: ")) - 1
26     bitterness = bitterness_options[b_choice]
27
28     print("\nВыберите желаемый аромат:")
29     for i, a in enumerate(aroma_options, 1):
30         print(f"{i}. {a}")
31     a_choice = int(input("Номер аромата: ")) - 1
32     aroma = aroma_options[a_choice]
33
34     print(f"\nПодбор по параметрам: горечь={bitterness},
35           аромат={aroma}")
36     matches = []
37     for name, data in beer_knowledge_base.items():
38         if data["bitterness"] == bitterness and data["aroma"] == aroma:
39             matches.append((name, data))
40
41     if matches:
42         print(f"\nНайдено {len(matches)} пива, соответствующего
43               вашим параметрам:")
44         for name, data in matches:
45             print(f"  {name}: {data['bitterness']} {data['aroma']}")
46     else:
47         print("Нет пива, соответствующего вашим параметрам")
```

```

34         if name == "Пиво":
35             continue
36         if data["горечь"] == bitterness and aroma in
37             data["аромат"]:
38             matches.append(name)
39     if matches:
40         print("Рекомендуется: ", ", ".join(matches))
41     else:
42         print("Не найдено подходящего типа пива.")
43
44 def find_by_country():
45     """Поиск пива по стране происхождения."""
46     # Собираем уникальные страны
47     countries = sorted(list({data["страна"] for name, data in
48         beer_knowledge_base.items() if name != "Пиво"}))
49
50     print("\nВыберите страну:")
51     for i, country in enumerate(countries, 1):
52         print(f"{i}. {country}")
53     choice = int(input("Номер страны: ")) - 1
54     if choice < 0 or choice >= len(countries):
55         print("Неверный выбор!")
56     return
57     country = countries[choice]
58
59     print(f"\n Пиво из страны: {country}")
60     result = [name for name, data in
61         beer_knowledge_base.items()
62             if name != "Пиво" and data["страна"] == country]
63     if result:
64         print("Найдено: ", ", ".join(result))
65     else:
66         print("Пиво из этой страны не найдено.")
67
68 def find_by_strength_and_color():
69     """Поиск по категории крепости и цвету."""
70     strengths = ["слабая", "средняя", "сильная"]
71     colors = ["светлый", "тёмный", "золотистый", "янтарный"]
72
73     print("\nВыберите крепость:")
74     for i, s in enumerate(strengths, 1):

```

```

72         print(f"{i}. {s}")
73 s_choice = int(input("Номер крепости: ")) - 1
74 strength = strengths[s_choice]
75
76     print("\nВыберите цвет:")
77     for i, c in enumerate(colors, 1):
78         print(f"{i}. {c}")
79     c_choice = int(input("Номер цвета: ")) - 1
80     color = colors[c_choice]
81
82     print(f"\n Пиво с крепостью '{strength}' и цветом
83           '{color}'")
84 result = []
85 for n, d in beer_knowledge_base.items():
86     if n == "Пиво":
87         continue
88     if d["крепость"] == strength and color.lower() in
89         d["цвет"].lower():
90         result.append(n)
91 if result:
92     print("Подходит:", ", ".join(result))
93 else:
94     print("Нет пива с такими характеристиками.")
95 def main():
96     while True:
97         print("\n==== Экспертная система: Вкусы пива ===")
98         print("1. Показать характеристики пива")
99         print("2. Рекомендовать пиво по вкусу")
100        print("3. Найти по стране происхождения")
101        print("4. Найти по крепости и цвету")
102        print("0. Выход")
103        choice = input("Выберите действие: ")
104
105        if choice == "1":
106            names = [n for n in beer_knowledge_base.keys() if
107                      n != "Пиво"]
108            print("\nВыберите пиво:")
109            for i, n in enumerate(names, 1):
110                print(f"{i}. {n}")
111            n_choice = int(input("Номер пива: ")) - 1
112            find_beer_by_name(names[n_choice])

```

```
110     elif choice == "2":  
111         recommend_beer()  
112     elif choice == "3":  
113         find_by_country()  
114     elif choice == "4":  
115         find_by_strength_and_color()  
116     elif choice == "0":  
117         print("Выход из системы")  
118         break  
119     else:  
120         print("Неверный выбор")  
121  
122 if __name__ == "__main__":  
123     main()
```

ПРИЛОЖЕНИЕ Б

Сформированная база данных

Листинг Б.1 – Исходный код программы

```
1 beer_knowledge_base = {
2     "Пиво": {
3         "тип": ["Лагер", "Старт", "Индийский светлый эль",
4                 "Пшеничное", "Портер", "Трипель",
5                 "Бельгийский Витбир", "Светлый Эль", "Кёльш",
6                 "Бок", "Пильзенер",
7                 "Американский Эль", "Овсяный Старт",
8                 "Барливайн", "Красный эль"],
9         "характеристики": ["горечь", "сладость", "аромат",
10                            "цвет", "крепость", "страна"]
11     },
12     "Лагер": {
13         "горечь": "низкая",
14         "сладость": "низкая",
15         "аромат": "хмелевой",
16         "цвет": "светлый",
17         "крепость": "слабая",
18         "страна": "Германия или Бельгия"
19     },
20     "Старт": {
21         "горечь": "высокая",
22         "сладость": "низкая",
23         "аромат": "кофейный",
24         "цвет": "тёмный",
25         "крепость": "сильная",
26         "страна": "Ирландия или Великобритания"
27     },
28     "Индийский светлый эль": {
29         "горечь": "высокая",
30         "сладость": "низкая",
31         "аромат": "хмелевой",
32         "цвет": "золотистый",
33         "крепость": "сильная",
34         "страна": "Великобритания"
35     },
36     "Пшеничное": {
37         "горечь": "низкая",
```

```
34     "сладость": "высокая",
35     "аромат": "разное",
36     "цвет": "светлый",
37     "крепость": "средняя",
38     "страна": "Германия или Бельгия"
39 },
40 "Портэр": {
41     "горечь": "средняя",
42     "сладость": "умеренная",
43     "аромат": "кофейный",
44     "цвет": "тёмный",
45     "крепость": "средняя",
46     "страна": "Великобритания"
47 },
48 "Трипель": {
49     "горечь": "средняя",
50     "сладость": "высокая",
51     "аромат": "пряный",
52     "цвет": "светлый",
53     "крепость": "сильная",
54     "страна": "Бельгия"
55 },
56 "Бельгийский Витбир": {
57     "горечь": "низкая",
58     "сладость": "умеренная",
59     "аромат": "пряный",
60     "цвет": "светлый",
61     "крепость": "слабая",
62     "страна": "Бельгия"
63 },
64 "Светлый Эль": {
65     "горечь": "низкая",
66     "сладость": "умеренная",
67     "аромат": "солодовый",
68     "цвет": "светлый",
69     "крепость": "средняя",
70     "страна": "Великобритания"
71 },
72 "Кёльш": {
73     "горечь": "низкая",
74     "сладость": "умеренная",
```

```
75     "аромат": "солодовый",
76     "цвет": "светлый",
77     "крепость": "средняя",
78     "страна": "Германия"
79 },
80 "Бок": {
81     "горечь": "средняя",
82     "сладость": "высокая",
83     "аромат": "карамельный",
84     "цвет": "тёмный",
85     "крепость": "средняя",
86     "страна": "Германия"
87 },
88 "Пильзенер": {
89     "горечь": "низкая",
90     "сладость": "низкая",
91     "аромат": "хмелевой",
92     "цвет": "светлый",
93     "крепость": "слабая",
94     "страна": "Чехия или Германия"
95 },
96 "Американский Эль": {
97     "горечь": "средняя",
98     "сладость": "умеренная",
99     "аромат": "солодовый",
100    "цвет": "золотистый",
101    "крепость": "средняя",
102    "страна": "США"
103 },
104 "Овсяный Старт": {
105    "горечь": "средняя",
106    "сладость": "высокая",
107    "аромат": "овсяный",
108    "цвет": "тёмный",
109    "крепость": "средняя",
110    "страна": "Ирландия"
111 },
112 "Барливайн": {
113    "горечь": "низкая",
114    "сладость": "высокая",
115    "аромат": "солодовый",
```

```
116     "цвет": "тёмный",
117     "крепость": "сильная",
118     "страна": "Бельгия или Германия"
119 },
120 "Красный эль": {
121     "горечь": "средняя",
122     "сладость": "умеренная",
123     "аромат": "карамельный",
124     "цвет": "красный",
125     "крепость": "средняя",
126     "страна": "Ирландия или Великобритания"
127 }
128 }
```