

## ДОСЛІДЖЕННЯ ПРИНЦИПІВ ПОБУДОВИ ЕКСПЕРТНИХ СИСТЕМ

**Мета роботи** – вивчити принципи побудови та навчитись розробляти експертні системи

### 1. Основні теоретичні відомості

Експертні системи можна поділити на два типи: засновані на фактах та засновані на правилах. Нижче наведено експертну систему на фактах, де знання про предметну галузь (у вигляді фактів) зберігаються у зовнішній базі даних. У разі зміни предметної галузі модифікується лише база даних, не змінюючи програму.

Розглянемо приклад експертної системи, яка дозволить за наявності певних ознак визначити вид риби. список ознак, які однозначно характеризують кожен з чотирьох видів риб, наведено у табл. 2.1.

*Таблиця 2.1*

Номер ознаки	Ознака	Значення ознаки	Сом	Плот ва	Окун ь	Щук а
1	Маса	> 20 кг	+			
2		< 20 кг	+	+	+	+
3	Тіло	Довге вузьке				+
4		Широке		+	+	
5	Луска	Звичайна			+	+
6		Срібляста		+		
7	Малюнок на тілі	Темні смуги			+	
8	Особливості	Має вуса	+			

Експертна система складається із сукупності модулів. База знань містить знання про предметну галузь. База даних (фактів) містить дані, введені користувачем. Блок інтерфейсу забезпечує діалог з користувачем. Механізм логічного виведення забезпечує пошук необхідних знань і фактів.

Опис предметної галузі зберігається в базі знань у вигляді двох предикатів:

1) ознаки та її номеру:

sign(numb\_s, stri), де numb\_s = integer, str = symbol;

2) риби, що розпізнається, та списку номерів ознак, які однозначно її характеризують:

`rule(fish, list_sign), де fish = symbol, list_sign = numb_s*.`

Тобто в секції `clauses` записуємо:

```
sign(1, маса більше 20 кг).  
sign(2, маса менше 20 кг).  
sign(3, тіло довге вузьке).  
sign(4, тіло широке).  
sign(5, луска звичайна).  
sign(6, луска срібляста).  
sign(7, темні смуги на тілі).  
sign(8, має вуси).  
rule(сом, [1, 2, 8]).  
rule(плотва, [2, 4, 6]).  
rule(окунь, [2, 4, 5, 7]).  
rule(щука, [2, 3, 5]).
```

Виконання програми починається запуском предиката `run` та з вибірки першого правила, яке містить назву тварини та список ознак `rule(Fish, List)`. Предикат `check(List)` перевіряє наявність ознак. Якщо всі ознаки наявні, то видається назва виду риби. Далі, у разі потреби, видається стан бази даних, тобто список відповідей користувача. Потім проводиться очищення бази даних від попередніх відповідей:

```
run:- rule(Fish, List),  
    heck(List), !,  
    write("Fish is ", Fish), nl,  
    status_db,  
    clear_facts.
```

Якщо розв'язку не знайдено, то видається повідомлення:

```
run:- write(«\n не можу визначити»), nl,  
    write(«\n що це за риба»), nl  
    status_db,  
    clear_facts.
```

Предикат контролю має рекурсивний характер. Обирається голова списку, перевіряється, чи немає в БД відомостей про поточну ознаку. В іншому випадку організується діалог з користувачем.

```
check([ ]). % якщо список порожній, то рекурсія припиняється  
check([Head| Rest]):-  
    yes(Head), !, % у базі даних є ця ознака
```

```

        check(Rest).                % перевірка хвоста списку
check([Head | _ ]):-
    no(Head), !, % у базі даних указується на відсутність ознаки
    fail.        % відкат до наступного правила
check([Head| Rest]):-
    sign(Head, Text),
    ask(Head, Text),
    check(Rest).                % перевірка хвоста списку

```

Організація діалогу з користувачем

```

ask(Head, Text):-
    write(«Запитання:»,Text, «yes/no?»), nl, % виведення запиту
                                           на екран
    readln(Reply), % запам'ятовування відповіді в Reply
    frontchar(Reply, First, _), % перша літера відповіді
    remember(Head, First). % запам'ятовування відповіді

```

У базі даних експертної системи спочатку немає ніяких фактів про поточну ситуацію. Тому в програму треба включати блок

операторів, який дозволить у ході діалогу з користувачем, отримувати інформацію і записувати її в базу даних:

```

database
    yes(numb_s)
    no(numb_s)

```

Відповіді будуть зберігатись у вигляді наявності або відсутності ознаки з відповідним номером.

Предикати поповнення бази даних:

```

remember(Head, 'y'):- assertz(yes(Head)).
remember(Head, 'n'):- assertz(no(Head)), fail.

```

%у випадку негативної відповіді – відкат

Предикат очищення бази даних після видачі відповіді:

```

clear_facts:-
    write("\n\n Для виходу натисніть будь-яку клавішу \n"),
    retractall( _ ),
    readchar( _ ).

```

Правило status\_db призначене для перевірки стану бази даних у вигляді списку відповідей користувачів на задані запитання:

```

status_db:- !,
    write(""),
    readln(Reply),
    frontchar(Reply, 'y', _ ),

```

```
print_positive,  
print_negative,  
write(" Завершення ")).
```

Виведення позитивних фактів з бази даних:

```
print_positive:-  
    write(" Позитивна відповідь " ), nl,  
    write("-----"), nl, fail.  
print_positive:-  
    xpositive(X,Y),  
    write(X, " ", Y),nl  
    fail.  
print_positive:-  
    write("-----"), nl.
```

Другий предикат необхідний, щоб підціль print\_positive була успішно доведена. Аналогічно виводяться відомості про негативні відповіді.

## **2. Порядок виконання роботи**

2.1. Скласти програму для експертної системи предметної галузі, заданої за варіантом.

2.2. Провести тестування експертної системи на декількох екземплярах предметної галузі.

2.3. Побудувати дерево виведення для одного екземпляра з предметної галузі.

**3. Звіт.** Звіт має містити таблицю ознак екземплярів предметної галузі, текст програми експертної системи, результати виконання за п.п. 2.2, 2.3.

## **4. Варіанти:**

1. Експертна система «Організація відпочинку».
2. Експертна система «Фрукти»
3. Експертна система «Автомобілі»
4. Експертна система «Квіти»
5. Експертна система «Тип темпераменту»
6. Експертна система «Предмети навчання»
7. Експертна система «Професія»
8. Експертна система «Пори року»
9. Експертна система «Птахи»
10. Експертна система «Риба»
11. Експертна система «Жанри музики»

12. Експертна система «Жанри фільмів»
13. Експертна система «Жанри книг»
14. Експертна система «Знаряддя праці»
15. Експертна система «Міста України»
16. Експертна система «Мова програмування»
17. Експертна система «Вид спорту»
18. Експертна система «Овочі»
19. Експертна система «Ягоди»
20. Експертна система «Гриби»

### **Контрольні запитання**

1. З яких модулів складається експертна система?
2. Які недоліки має продукційна експертна система?
3. У чому відмінність динамічної експертної системи від статичної?
4. Як зберігаються знання в експертній системі на предикатах?
5. Для чого необхідно записувати відповіді користувача у базу даних?
6. Які предикати використовуються для додавання та видалення фактів у базі даних?