

Функциональное и логическое программирование

Лекция 8

2.14 Предикаты для работы с файлами (продолжение)

Пример 2:

Написать предикат, который выводит на экран строки из файла, начиная с некоторого номера. Имя файла и номер строки вводятся с клавиатуры.

```
goal1:-writeln('имя файла? '),read(Filename),
        check(Filename),
        writeln('Номер строки для вывода? '),read(N),
        open(Filename,read,F),
        set_input(F),
        read_file(F,N),
        format('Содержимое файла с ~w-ой строки\n',N),
        write_file(F),
        close(F).

check(Filename):-exists_file(Filename),!.
check(Filename):-format('Файла с именем ~w не существует\n',Filename),fail.
read_file(_,N):-at_end_of_stream,!,
                format('В файле меньше чем ~w строк\n',N),fail.
read_file(_,1):-!.
read_file(F,N):-read_line_to_codes(F,_),
                N1 is N-1,
                read_file(F,N1).
write_file(_):-at_end_of_stream,!.
write_file(F):-read_line_to_codes(F,L),
                string_to_list(S,L),
                writeln(S),
                write_file(F).
```

?- goal1.

имя файла?

|: 'in.txt'.

Номер строки для вывода?

|: 6.

Содержимое файла с 6-ой строки

But these are transient all;

If the shower will make the roses bloom,

Oh, why lament its fall?

Rapidly, merrily,

Life's sunny hours flit by,

Gratefully, cheerily,

Enjoy them as they fly!

true.

Содержимое файла in.txt:

Life, believe, is not a dream

So dark as sages say;

Oft a little morning rain

Foretells a pleasant day.

Sometimes there are clouds of gloom,

But these are transient all;

If the shower will make the roses bloom,

Oh, why lament its fall?

Rapidly, merrily,

Life's sunny hours flit by,

Gratefully, cheerily,

Enjoy them as they fly!

Пример 3:

Написать предикат, который записывает вводимые с клавиатуры строки в файл out.txt. Окончание ввода – строка “#”.

```
goal2:-writeln('Введите строку для вывода в файл '),
        read(S),
        tell('out.txt'),
        write_s(S),
        told.
write_s("#"):-!.
write_s(S):-writeln(S),
              read(S1),
              write_s(S1).
```

Содержимое файла out.txt:

```
dfg ghj kkk ,
123 dff ggh
```

```
?- goal2.
Введите строку для вывода в файл
|: "dfg ghj kkk ,".
|: "123 dff ggh ".
|: "#".

true.
```

2.15 Динамические базы данных

Программа – реляционная база данных. В процессе работы может возникнуть необходимость изменить, удалить, добавить предложения. Такие предложения – часть динамической базы данных.

Директива

`:-dynamic <имя предиката>/<арность>.`

Если динамическими являются несколько предикатов, то они перечисляются через запятую.

2.15.1 Добавление и удаление предложений

`asserta(<предложение>)`

Добавление в начало базы данных.

`assertz(<предложение>)`

Добавление в конец базы данных.

`assert(<предложение>)`

Добавление в конец базы данных.

Предикаты, добавляемые с помощью `asserta` и `assertz`, становятся динамическими по умолчанию.

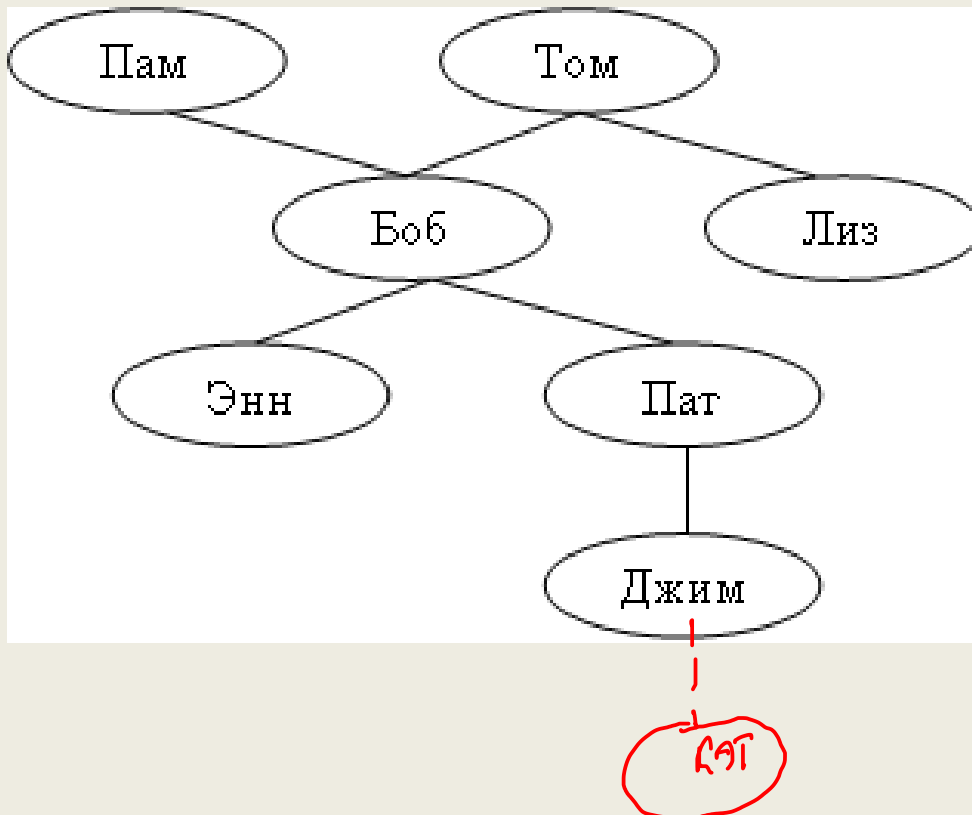
Пример 1: Пусть имеется дерево семейных отношений, описанное программой в лекции 4:

родитель(пам,боб).
родитель(том,боб).
родитель(том,лиз).
родитель(боб,энн).
родитель(боб,пат).
родитель(пат,джим).
мужчина(том).
мужчина(боб).
мужчина(джим).

?- assert(родитель(джим,кэт)).
true.

?- родитель(джим,Я).
Я = кэт.

`:-dynamic родитель/2.`



`retract(F)`

Удаление из динамической базы данных первого предложения, сопоставимого с F. Предикат всегда успешен.

`retractall(F)`

Удаление из динамической базы данных всех предложений, сопоставимых с F. Предикат всегда успешен.

Продолжим работу после добавления в динамическую базу данных факта

```
?- retract(родитель(джим,кэт)).  
true.
```

```
?- родитель(джим,Я).  
false.
```


В динамическую базу можно добавлять правила:

```
?- assert((дед(X,Y):-мужчина(X),родитель(X,Z),родитель(Z,Y))).  
true.
```

```
?- дед(Ч,джим).  
Ч = боб  
Unknown action: ж (h for help)  
Action? ;  
false.
```

```
?- retractall(дед(_,_)).  
true.
```

```
?- дед(Ч,джим).  
false.
```

Предложения, добавленные к программе приведенным выше способом, ведут себя точно так же, как и те, что были в “оригинале” программы.

listing(<имя предиката>/<арность>)

Вывод всех предложений базы данных, относящихся к определенному предикату, в текущий выходной поток.

```
?- assert((дед(X,Y):-мужчина(X),родитель(X,Z),родитель(Z,Y))).  
true.
```

```
?- listing(дед/2).  
:- dynamic дед/2.
```

```
дед(A, B) :-  
    мужчина(A),  
    родитель(A, C),  
    родитель(C, B).
```

```
true.
```

Предикат `listing` без аргументов выводит все содержимое базы данных. В том числе, там хранится и код программы (в конце базы данных).

2.15.2 Заполнение динамической базы данных из файла, сохранение в файл

`consult('<имя файла>')`

Считывает из файла предложения и добавляет их в конец динамической базы данных (аналогично `assert`).

Сохранение из динамической базы данных в файл:
`tell+listing.`

Пример 3:

Формирование динамической базы данных «Читатель библиотеки» с клавиатуры и сохранение ее в файле reader.txt.

```
goal3:-repeat,  
    writeln('будете вводить новые данные y/n? '), read(Answer),  
    ответ(Answer),!,  
    tell('reader.txt'),  
    listing(читатель/2),  
    retractall(читатель(_, _)),  
    told.  
ответ(n).  
ответ(y):-запись,fail.  
ответ(_):-fail.  
запись:-writeln('ФИО ? '), read(F),  
    writeln('Месяц посещения ? '), read(M),  
    writeln('Число посещения ? '), read(D),  
    assert(читатель(F,дата_посещения(D,M))).
```

?- goal3.

будете вводить новые данные у/n?

|: a.

будете вводить новые данные у/n?

|: у.

ФИО ?

|: a.

Месяц посещения ?

|: 6.

Число посещения ?

|: 20.

будете вводить новые данные у/n?

|: у.

ФИО ?

|: b.

Месяц посещения ?

|: 8.

Число посещения ?

|: 1.

будете вводить новые данные у/n?

|: r.

будете вводить новые данные у/n?

|: n.

true.

Содержимое файла reader.txt:

:- dynamic читатель/2.

читатель(a, дата_посещения(20, 6)).

читатель(b, дата_посещения(1, 8)).

Пример 4:

Определение количества читателей, посетивших библиотеку в мае, по информации, находящейся в файле reader.txt, сформированном в примере 3.

```
:-dynamic счетчик/1.  
goal4:-consult('reader.txt'),  
    assert(счетчик(0)),  
    счет,счетчик(N),  
    format('Кол-во читателей в мае ~w ',N),  
    retractall(счетчик(_)).  
счет:-читатель(_,дата_посещения(_,5)),  
    счетчик(N), N1 is N+1,  
    retract(счетчик(N)),assert(счетчик(N1)),  
    fail.  
счет.
```

```
?- goal4.  
Кол-во читателей в мае 4  
true.
```

Содержимое файла reader.txt:

```
:- dynamic читатель/2.  
  
читатель(a, дата_посещения(20, 6)).  
читатель(b, дата_посещения(1, 8)).  
читатель(c, дата_посещения(1, 5)).  
читатель(d, дата_посещения(1, 5)).  
читатель(e, дата_посещения(1, 7)).  
читатель(f, дата_посещения(1, 5)).  
читатель(g, дата_посещения(1, 5)).
```

2.16 Создание меню

Пример:

Напишем предикат `menu`, который создает окно с главным меню, состоящим из трех пунктов. Выбор пунктов главного меню происходит до тех пор, пока не будет выбран 3 пункт (выход).

```
menu:-repeat,  
    writeln('1 - процесс 1'),  
    writeln('2 - процесс 2'),  
    writeln('3 - выход'),  
    writeln('выберите пункт меню'),  
    read(N),  
    N<4,  
    process(N),  
    N=3,!.  
process(3).  
process(1):-writeln('отработал процесс 1'),fail.  
process(2):-writeln('отработал процесс 2'),fail.
```

```
?- menu.  
1 - процесс 1  
2 - процесс 2  
3 - выход  
выберите пункт меню  
|: 1.  
отработал процесс 1  
1 - процесс 1  
2 - процесс 2  
3 - выход  
выберите пункт меню  
|: 2.  
отработал процесс 2  
1 - процесс 1  
2 - процесс 2  
3 - выход  
выберите пункт меню  
|: 6.  
1 - процесс 1  
2 - процесс 2  
3 - выход  
выберите пункт меню  
|: 3.  
  
true.
```