



КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Преподаватель Толпинская Н.Б., Строганов Ю. В.

Лабораторная работа №18

Постановка задачи

Используя хвостовую рекурсию, разработать программу, позволяющую найти:

1. $n!$;
2. n -е число Фибоначчи.

Решение

```
1 domains
2   num = integer
3
4 predicates
5   fact(num, num)
6   rec_fact(num, num, num)
7
8   fib(num, num)
9   rec_fib(num, num, num, num)
10
11 clauses
12   rec_fact(N, Res, Acc) :- N > 1, !, NewN = N - 1, NewAcc = Acc * N,
13     rec_fact(NewN, Res, NewAcc).
14   rec_fact(_, Res, Acc) :- Res = Acc.
15   fact(N, Res) :- rec_fact(N, Res, 1).
16
17   rec_fib(N, F1, F2, Res) :- N > 2, !, NewF1 = F2, NewF2 = F1 + F2, NewN = N
18     - 1, rec_fib(NewN, NewF1, NewF2, Res).
19   rec_fib(_, _, B, Res) :- Res = B.
20   fib(N, Res) :- rec_fib(N, 1, 1, Res).
21
22 goal
23   %fact(4, Res).
24   fib(7, Res).
```

Лабораторная работа №19

Постановка задачи

Задание: используя хвостовую рекурсию, разработать эффективную программу (комментируя назначение аргументов), позволяющую:

1. Найти длину списка (по верхнему уровню);
2. Найти сумму элементов числового списка;
3. Найти сумму элементов числового списка, стоящих на нечетных позициях исходного списка (нумерация от 0);

Решение

```
1 domains
2   intlist = integer*
3
4 predicates
5   rec_length(integer, integer, intlist)
6   length(integer, intlist)
7
8   rec_sum(integer, integer, intlist)
9   sum(integer, intlist)
10
11  rec_oddsum(integer, integer, intlist)
12  oddsum(integer, intlist)
13
14 clauses
15  rec_length(Res, Len, [_ | Tail]) :- NewLen = Len + 1, !, rec_length(Res,
16    NewLen, Tail).
17  rec_length(Res, Len, []) :- Res = Len.
18  length(Res, List) :- rec_length(Res, 0, List).
19
20  rec_sum(Res, Sum, [Head | Tail]) :- NewSum = Sum + Head, !, rec_sum(Res,
21    NewSum, Tail).
22  rec_sum(Res, Sum, []) :- Res = Sum.
23  sum(Res, List) :- rec_sum(Res, 0, List).
```

```
23  rec_oddsum(Res, Sum, [_ , Head | Tail]) :- NewSum = Sum + Head, !,  
    rec_oddsum(Res, NewSum, Tail).  
24  rec_oddsum(Res, Sum, []) :- Res = Sum.  
25  oddsum(Res, List) :- rec_oddsum(Res, 0, List).  
26  
27  goal  
28  %length(Res, [1, 2, 3, 4]).  
29  %sum(Res, [1, 2, 3, 4]).  
30  oddsum(Res, [1, 2, 3, 4]).
```

Лабораторная работа №20

Постановка задачи

Задание: используя хвостовую рекурсию, разработать, комментируя аргументы, эффективную программу, позволяющую:

1. Сформировать список из элементов числового списка, больших заданного значения;
2. Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);
3. Удалить заданный элемент из списка (один или все вхождения);
4. Преобразовать список в множество (можно использовать ранее разработанные процедуры).

Решение

```
1 domains
2   intlist = integer*
3
4 predicates
5   bigger_than(intlist, integer, intlist)
6   odd_list(intlist, intlist)
7   single_del(intlist, integer, intlist)
8   full_del(intlist, integer, intlist)
9   set(intlist, intlist)
10
11 clauses
12   bigger_than([Head | Tail], N, [Head | ResTail]) :- Head > N, !,
13       bigger_than(Tail, N, ResTail).
14   bigger_than([_ | Tail], N, Result) :- bigger_than(Tail, N, Result).
15   bigger_than([], _, []).
16
17   odd_list([_, Head | Tail], [Head | ResTail]) :- !, odd_list(Tail, ResTail)
18   .
19   odd_list([], []).
20
21   single_del([Head | Tail], N, Tail) :- Head = N, !.
```

```

20 single_del([Head | Tail], N, [Head | ResTail]) :- single_del(Tail, N,
    ResTail), !.
21 single_del([], _, []).
22
23 full_del([Head | Tail], N, [Head | ResTail]) :- Head <> N, !, full_del(
    Tail, N, ResTail).
24 full_del([_ | Tail], N, Result) :- full_del(Tail, N, Result), !.
25 full_del([], _, []).
26
27 set([Head | Tail], [Head | Result]) :- full_del(Tail, Head, Nt), !, set(Nt
    , Result).
28 set([], []).
29
30 goal
31 %bigger_than([1, 7, 3, 4, 5, 6], 3, Result).
32 %odd_list([1, 2, 3, 4, 5, 6, 7, 8], Result).
33
34 %single_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
35 %full_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
36
37 set([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).

```