Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №4

РЕКУРСИВНАЯ ОБРАБОТКА ДАННЫХ

по дисциплине «Логическое программирование»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Ткаченко К.С.

г. Севастополь 2019

**Цель работы**

Изучить основные режимы работы интерфейса Visual Prolog, структуру программы и ее разделы. Ознакомиться с понятием «рекурсия».

**Вариант № 23(23, 7, 21)**

Задание:

Выполнить задание согласно вариантам.

* Вариант 23 - Построить рекурсивную функцию для вычисления n-го члена последовательности, в которой каждый следующий член равен произведению двух предыдущих, а первые 2 члена равны соответственно 1 и 2.
* Вариант 7 - Вычислить: Y = k\*(k+1)\*(k+2)\*….\*(k+n).
* Вариант 21 - Построить рекурсивную функция для вычисления n-го члена последовательности, в которой каждый четный член равен сумме двух предыдущих четных, а нечетный равен сумме двух предыдущих нечетных. Первые четыре члена равны соответственно 1, 2, 3, 4.

**Ход работы**

**Задание по варианту 23.**

Программа нахождения n-го члена последовательности, в которой каждый следующий член равен произведению двух предыдущих, а первые 2 члена равны соответственно 1 и 2.

Дерево И/ИЛИ:

f(N,F)

N = 1

N = 2

F = 2

F = 1

!

F = F1 \* F2

f(N-2,F2)

!

f(N-1,F1)

Текст программы:

implement main  
    open core  
  
clauses            
 classinfo("L2", "4.0").    
   
f(N,F) :-   
        N = 1, !, F = 1;  
        N = 2, !, F = 2;  
        f(N-1,F1), f(N-2,F2), F = F1 \* F2.  
 loop(N) :-  
        ( N = 1, !, f(1,F);  
          loop(N-1), f(N,F) ),  
        stdio::write(F, "  ").  
  
clauses  
run():-  
    console::init(),  
    stdio::write("Введите N"),  
    stdio::nl,  
    loop(stdio::read()),   
    succeed().  
end implement main  
     
goal        
   mainExe::run(main::run).

Результат выполнения программы:



**Задание по варианту 7.**

Программа, которая вычисляет: Y = k\*(k+1)\*(k+2)\*….\*(k+n).

Дерево И/ИЛИ:

f(K,F,N)

N = 0

F = K

!

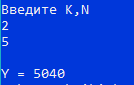
F = (K+N) \* F1

f(K,F1,N-1)

Текст программы:

implement main  
    open core  
  
clauses            
 classinfo("L2", "4.0").    
   
f(K,F,N) :-   
        N = 0, !, F = K;  
        f(K,F1,N-1), F = (K+N) \* F1.  
  
clauses  
run():-  
    console::init(),  
    stdio::write("Введите K,N"),  
    stdio::nl,  
    f(stdio::read(),Y,stdio::read()),   
stdio::nl,  
    stdio::write("Y = ", Y),  
    succeed().  
end implement main  
     
goal        
   mainExe::run(main::run).

Результат выполнения программы:



**Задание по варианту 21.**

Программа, которая вычисляет n-ый член последовательности, в которой каждый четный член равен сумме двух предыдущих четных, а нечетный равен сумме двух предыдущих нечетных. Первые четыре члена равны соответственно 1, 2, 3, 4.

Дерево И/ИЛИ:

f(N,F)

N = 1

N = 2

F = 2

F = 1

!

F = F1 + F2

f(N-2,F2)

!

f(N-4,F1)

N = 3

!

F = 3

N = 1

F = 1

!

Текст программы:

implement main  
    open core  
  
clauses            
 classinfo("L2", "4.0").    
   
f(N,F) :-   
        N = 1, !, F = 1;  
        N = 2, !, F = 2;  
        N = 3, !, F = 3;  
        N = 4, !, F = 4;  
        f(N-4,F1), f(N-2,F2), F = F1 + F2.  
 loop(N) :-  
        ( N = 1, !, f(1,F);  
          loop(N-1), f(N,F) ),  
        stdio::write(F, "  ").  
  
clauses  
run():-  
    console::init(),  
    stdio::write("Введите N"),  
    stdio::nl,  
    loop(stdio::read()),   
    succeed().  
end implement main  
     
goal        
   mainExe::run(main::run).

Результат выполнения программы:



**Выводы**

Вданной лабораторной работе были изучены основные режимы работы интерфейса VisualProlog, структура программы и ее разделы. Были получены знания о понятии «рекурсия».