**Лабораторная работа №5**

**Вариант №11**

**Выполнил: Котолевский М.Н. 19 группа.**

Рекурсивные структуры данных (деревья)

**Цель работы:**

Соберите узлы заданного уровня в список.

Узел бинарного дерева находится на уровне N, если путь от родительского узла до данного узла имеет длину N-1. Родительский элемент находится на уровне 1. Напишите метод atlevel/3 для сборки всех узлов выбранного уровня в список.

**Ход работы:**

Составим программу, которая удовлетворяет требованиям цели работы.

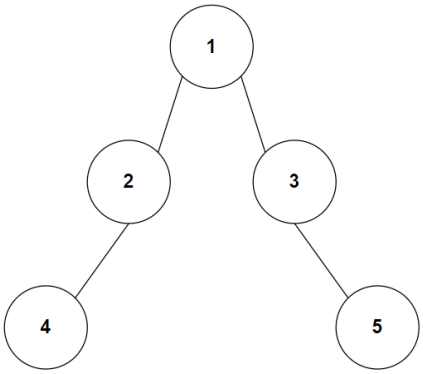
Предикат levelorder реализует обход дерева в порядке уровней, вызывая предикат atlevel для каждого уровня и объединяя все списки узлов в один список. Предикат atlevel рекурсивно проходит по левому и правому поддеревьям и находит узлы на заданном уровне, добавляя их в список.

Таким образом, levelorder поочередно вызывает atlevel для каждого уровня дерева, чтобы получить список узлов в порядке уровней.

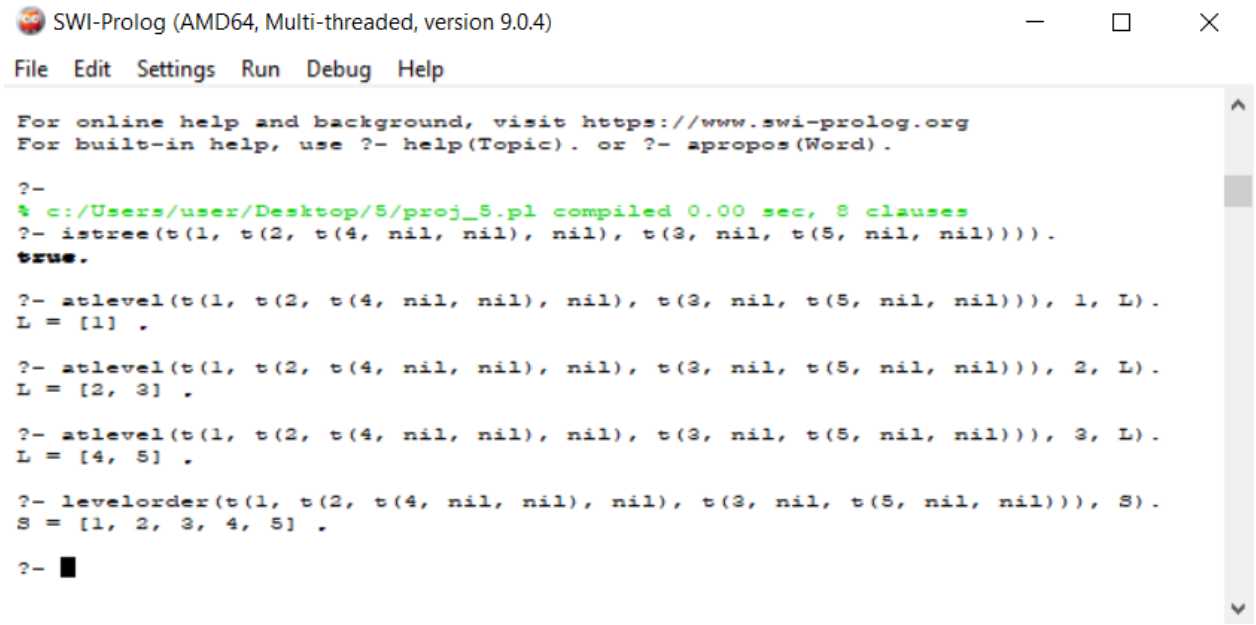
```
istree(nil).

istree(t(_,L,R)) :- istree(L), istree(R).
%определяет, что дерево t(_,L,R) является деревом, если его левое поддерево L и правое поддерево R также являются деревьями.

atlevel(nil,_,[]).
%определяет, что на любом уровне дерева nil нет ни одного узла, и возвращает пустой список [].

atlevel(t(X,_,_),1,[X]).
%определяет, что на первом уровне дерева t(X,_,_) есть только один узел X, и возвращает список [X].

atlevel(t(_,L,R),D,S) :- D > 1, D1 is D-1, atlevel(L,D1,SL), atlevel(R,D1,SR), append(SL,SR,S).
%определяет, что на уровне D дерева t(_,L,R) находятся узлы из списков SL и SR, соответственно, для левого и правого поддеревьев L и R, и объединяет эти списки в один список S.

levelorder(T,S) :- levelorder(T,S,1).
%определяет последовательность всех узлов бинарного дерева T в порядке обхода дерева по уровням, и возвращает эту последовательность в списке S.

levelorder(T,[],D) :- atlevel(T,D,[]), !.
%определяет, что на уровне D дерева T нет ни одного узла, и возвращает пустой список [].
```

levelorder(T,S,D) :- atlevel(T,D,SD), D1 is D+1, levelorder(T,S1,D1), append(SD,S1,S).
%определяет, что на уровне D дерева T находятся узлы из списка SD, и продолжает обход дерева на следующем уровне D1 = D+1, вызывая рекурсивно levelorder(T,S1,D1), и затем объединяет списки SD и S1 в список S. Когда уровень D становится больше, чем высота дерева, рекурсия останавливается, и последний вызов levelorder(T,[],D) возвращает пустой список [].

## Дерево:




## Результаты работы:




## Запросы:

istree(t(1, t(2, t(4, nil, nil), nil), t(3, nil, t(5, nil, nil)))).

atlevel(t(1, t(2, t(4, nil, nil), nil), t(3, nil, t(5, nil, nil))), 1, L).

atlevel(t(1, t(2, t(4, nil, nil), nil), t(3, nil, t(5, nil, nil))), 2, L).

atlevel(t(1, t(2, t(4, nil, nil), nil), t(3, nil, t(5, nil, nil))), 3, L).

levelorder(t(1, t(2, t(4, nil, nil), nil), t(3, nil, t(5, nil, nil))), S).

## Вывод:

Создана в системе программа, которая решает задачи согласно варианту. Изучена работа с деревьями.