

# Зертханалық жұмыс N°10

Бинарлық бұтақтар



Деректер құрылымнда рекурсияның болуын қабылдайтын деректер типі рекурсивті болып табылады.

Рекурсивті деректер типіне *бинарлық бұтақтар* мысал бола алады. Бинарлық бұтақтар не бос болады, не келесі үш бөлімнен тұрады:

түбір;

сол жақ ішкі бұтақ;

оң жақ ішкі бұтақ.

Мұндағы сол жақтағы және оң жақтағы ішкі бұтақтар өздері бинарлық бұтақтар болып табылады. Сонымен, бинарлық бұтақтың рекурсивтілігі оның анықтамасының өзінде көрініп тұр.

Бинарлық бұтақ реттелген деп есептеледі, егер оның сол жақ ішкі бұтақтарының барлық төбелері түбірден кіші болса, ал оң жақ ішкі бұтақтарының барлық төбелері түбірден үлкен болса және екі ішкі бұтақтары да реттелген болса. Мұндай бұтақ бинарлық анықтамалық деп аталады. Реттеудің артықшылығы мынада, бинарлық анықтамалықтан қандай-да бір объектіні табу үшін, бір ғана ішкі бұтақты қарап шығу жеткілікті.

Бинарлық бұтақты құруға және модификациялауға арналған предикаттарды қолданаймыз. Бинарлық бұтақ келесі функтор көмегімен беріледі.



`tree(K, LeftT, RightT),`

где  $K$  – элемент, находящийся в вершине; `LeftT` и `RightT` – левое и правое поддерево соответственно.

`create_tree(A, tree(A, empty, empty)).` %

создание дерева

`insert_left(X, tree(A, _, B), tree(A, X, B)).` % включение  
элемента данных  $A$ , как левого поддерева  $B$

`insert_right(X, tree(A, B, _), tree(A, B, X)).` %  
включение элемента данных  $A$ , как правого поддерева  $B$

Для обхода бинарного дерева «сверху вниз» опишем предикат:

`up_to_down(tree(X, LTr, RTr), Xs) :- up_to_down(LTr, Ls), up_to_down(RTr, Rs), append([X|Ls], Rs, Xs).`

`up_to_down(empty, []).`

`append` – это процедура `append(LeftList, RightList, ListRes)`, где `ListRes` является результатом слияния списков `LeftList`, `RightList`.



## Бақылау мысалы

Программа бөлігі, оның тереңдігінен өтіп, барлық элементтерін баспаға шығарады.

```
show_tree(nil).
```

```
show_tree(tree(X,Left,Right)):-write(X),show_tree(Left),  
show_tree(Right).
```

