

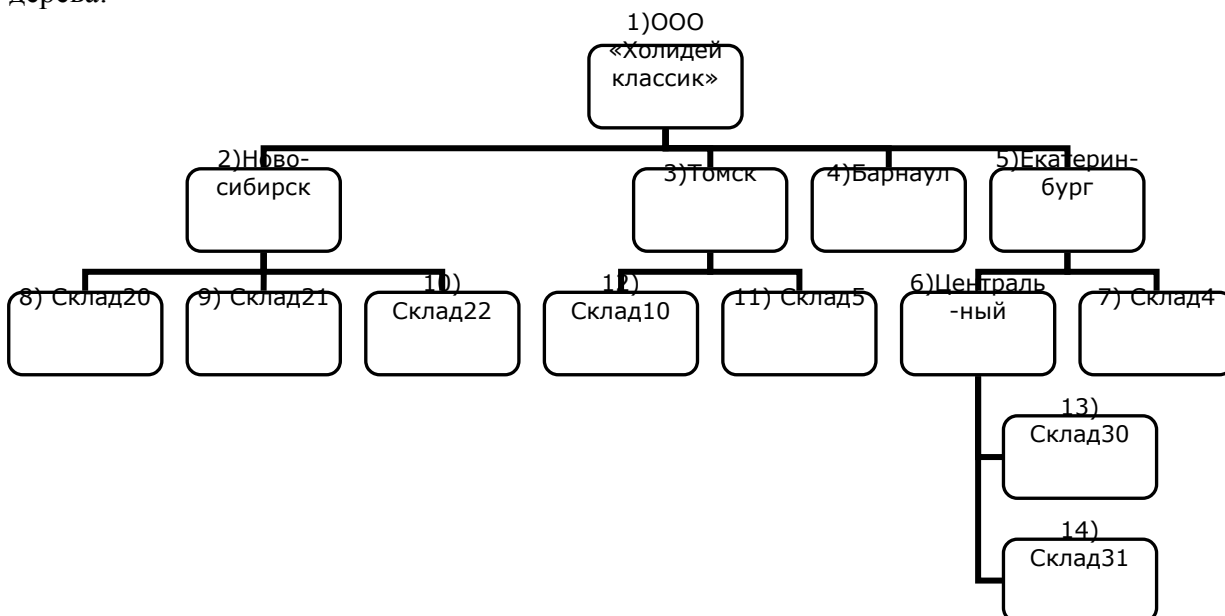
## Работа с иерархическими структурами данных. Операции на деревьях.

Дерево-структура данных, представляющих собой ориентированный связный граф. Элементы данных, составляющие дерево, обычно называют «узлами». Для обозначения коллекции узлов (множества элементов, составляющих дерево) используют термин *nodes*. Для обозначения *i*-го элемента коллекции узлов используют ссылку вида *nodes(i)*. Каждый элемент в такой структуре может иметь только один родительский элемент (*parent node*) и может иметь один или несколько дочерних элементов (*child nodes*). В каждом дереве имеется как минимум один особый элемент, у которого нет родительского узла. Такой элемент называется корнем дерева либо корневым узлом (*root node*). Каждый узел должен иметь свой уникальный ключ и реквизиты. Минимальный набор реквизитов - один реквизит - подпись узла, обычно используют два реквизита – подпись узла и числовое значение узла (некоторая сумма, характеризующая узловой объект).

Примеры древовидных структур:

- файловая система. Логические диски представляют собой корни, папки и файлы представляют собой узлы такого дерева. Если в папке содержатся файлы и вложенные папки, то для этой папки они являются дочерними узлами. Для этого дерева атрибутом будет являться имя папки или файла. Ключевым значением является физический адрес элемента файловой системы.
- иерархическая структура организации или предприятия. Корневым узлом будет головная организация. Дочерними узлами могут быть филиалы, отделы. В качестве ключевого значения может быть выбран ИНН, в качестве реквизитов может быть выбран доход или выручка в зависимости от поставленной задачи.

Например, структуру организации ООО «Холидей классик» можно представить в виде дерева:



Представим эту же структуру в виде таблицы. Каждая строка таблицы соответствует узлу. Номер элемента разместим в колонке *Key*, код родительского элемента – в колонке *RKey*, наименование узла в колонке *Name*, для хранения числовой характеристики узла добавим колонку *Summa*

Key	PKey	Naimen	Summa
1		ООО «Холидей классик»	17000
2	1	Новосибирск	34000
3	1	Томск	5000
4	1	Барнаул	3000
5	1	Екатеринбург	4000
6	5	Центральный	1000
7	5	Склад4	2000
8	2	Склад20	8200
9	2	Склад21	2000
10	2	Склад22	3500
11	3	Склад5	125000
12	3	Склад10	40500
13	6	Склад30	35000
14	6	Склад31	85000

При использовании подобных структур в программах, можно хранить соответствующую таблицу в виде массива, текстового файла, dbf-таблицы либо таблицы какой-либо другой СУБД.

В нижеследующих примерах, для удобства, будет использовано представление в виде dbf-таблицы – derevo.dbf.

Структура dbf-таблицы:

derevo(key I, PKey , Naimen C(50), Summa\_ Y).

Здесь key – уникальный ключ каждого элемента структуры, pkey – родительский ключ, naimen – наименование организации и ее структурных подразделений, Summa\_ - суммы соответствующих подразделений. Использование dbf-таблицы дает возможность использовать команду поиска locate for с функцией found(), команду CALCULATE FOR to с агрегатными функциями: sum(), count(), avg(), min(), max() и т.п.

## **Примеры процедур для работы с древовидными структурами на языке VFP.**

Типичные задачи для работы с древовидными структурами:

- Получение количества корневых элементов
- Получение количества дочерних узлов
- Получение ключа корневого элемента по порядковому номеру
- Получение ключа дочернего элемента по порядковому номеру
- Добавление дочерних узлов
- Получение суммы узла по ключу
- Сумма дочерних узлов
- Сумма дочерних узлов с заменой.
- Если у узла есть дочерние узлы, то для всех дочерних узлов сумму запомнить в новую переменную
- Заменить значение в узле с родительским кодом на новую сумму, состоящую из суммы всех дочерних узлов
- Если дочерних узлов нет, оставить сумму родительского узла без изменения
- Заменять сумму в узле суммой дочерних узлов
- Получение по ключу названия узла

## 1. Получение названия по ключу

В процедуре получения названия по ключу передадим как параметр родительский ключ в переменную lp\_key

```
LPARAMETERS lp_key
```

Сделаем проверку открыта ли таблица, если нет, то откроем ее

```
IF NOT USED("derevo")
```

```
USE derevo IN 0
```

```
ENDIF
```

Командой

```
SELECT derevo
```

Выбираем таблицу

```
locate FOR lp_key==ALLTRIM(key)
```

если находим

```
IF FOUND()
```

Возвращаем наименование

```
RETURN naimen
```

иначе

```
ELSE
```

Возвращаем

```
RETURN "нет такого узла"
```

```
ENDIF
```

## 2. Замена значения, имени узла по ключу

```
CLEAR
```

```
SET DEFAULT TO "u:\tree"
```

```
IF замена_текста("38_", "facultet")
```

```
? "замена сделана"
```

```
ELSE
```

```
? "замены нет"
```

```
ENDIF
```

```
IF замена_суммы("20_", 20000)
```

```
? "замена сделана"
```

```
ELSE
```

```
? "замены нет"
```

```
ENDIF
```

```
*добавление_дочерних_узлов("20_", "org0", 30000)
```

```
*!*    добавление_дочерних_узлов("20_", "org1", 50000)
```

```
*!*    добавление_дочерних_узлов("20_", "org2", 80000)
```

```
*!*    добавление_дочерних_узлов("20_", "org3", 50000)
```

```
? "сумма_дочерних_узлов_с_заменой для узла 16_=",  
сумма_дочерних_узлов_с_заменой("16_")
```

```
? "сумма_дочерних_узлов_с_заменой для узла 45_=",  
сумма_дочерних_узлов_с_заменой("45_")
```

```

kor=получение_количества_корневых_элементов()
?"корневых элементов=", kor
FOR i=1 TO kor
x=получение_ключа_корневого_элемента_по_порядковому_номеру(i)
?"корневой элемент номер=", i, "имеет ключ", x
?"дочерних элементов=", получение_количества_дочерних_узлов(x)

    FOR k=1 TO получение_количества_дочерних_узлов(x)
    ? k, получение_ключа_дочернего_элемента_по_порядковому_номеру(x,k)

    ENDFOR
ENDFOR

```

```

k="2_"
a=получение_названия_по_ключу(k)
s=получение_суммы_узла_по_ключу(k)
n=получение_количества_дочерних_узлов(k)
?"Для ключа=",k," Наименование=",a," сумма=", s, n

```

```

k="17_"
a=получение_названия_по_ключу(k)
s=получение_суммы_узла_по_ключу(k)
n=получение_количества_дочерних_узлов(k)

?"Для ключа=",k," Наименование=",a," сумма=", s, n

```

```

k="31_"
a=получение_названия_по_ключу(k)
s=получение_суммы_узла_по_ключу(k)
n=получение_количества_дочерних_узлов(k)

?"Для ключа=",k," Наименование=",a," сумма=", s, n

```

```

?"сумма_дочерних_узлов=", сумма_дочерних_узлов("16_")

```

```

PROCEDURE получение_названия_по_ключу
LPARAMETERS lp_key
IF NOT USED("derevo")
USE derevo IN 0
ENDIF

```

```

SELECT derevo
locate FOR lp_key==ALLTRIM(key)
IF FOUND()
RETURN naimen
ELSE
RETURN "нет такого узла"
ENDIF

```

```

PROCEDURE получение_суммы_узла_по_ключу
LPARAMETERS lp_key
IF NOT USED("derevo")

```

```
USE derevo IN 0
ENDIF
```

```
SELECT derevo
locate FOR lp_key==ALLTRIM(key)
  IF FOUND()
RETURN summa_
ELSE
RETURN 0
ENDIF
```

```
PROCEDURE получение_количества_дочерних_узлов
LPARAMETERS lp_key
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
```

```
SELECT derevo
COUNT FOR lp_key==ALLTRIM(pkey) TO b
RETURN b
```

```
PROCEDURE получение_количества_корневых_элементов
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
```

```
SELECT derevo
COUNT FOR LEN(ALLTRIM(pkey))=0 TO g
RETURN g
```

```
PROCEDURE получение_ключа_корневого_элемента_по_порядковому_номеру
LPARAMETERS n
LOCAL i
* порядковый номер корневого элемента
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
i=0
SELECT derevo
SCAN

IF LEN(ALLTRIM(pkey))=0
i=i+1
  IF i=n
RETURN ALLTRIM(key)
  ENDIF
ENDIF
ENDIF
```

```
ENDSCAN
return "не найдено"
```

```

PROCEDURE получение_ключа_дочернего_элемента_по_порядковому_номеру
LPARAMETERS bas_key,n
LOCAL i
* bas_key-ключ базового элемента, n-порядковый номер корневого элемента
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
i=0
SELECT derevo
SCAN

IF ALLTRIM(pkey)=bas_key
i=i+1
  IF i=n
    RETURN ALLTRIM(key)
  ENDIF
ENDIF

ENDSCAN
return "не найдено"

```

```

PROCEDURE замена_текста
LPARAMETERS key_, naimen_
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
SELECT derevo
LOCATE FOR key_=ALLTRIM(key)
IF FOUND()
replace naimen WITH naimen_
RETURN .t.
ELSE
return .f.
ENDIF

```

```

PROCEDURE замена_суммы
LPARAMETERS key_, sum_
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
SELECT derevo
LOCATE FOR key_=ALLTRIM(key)
IF FOUND()
replace summa_ WITH sum_
RETURN .t.
ELSE
return .f.
ENDIF

```

```

PROCEDURE добавление_дочерних_узлов
lparameters bas_key, naimen_, sum_
LOCAL s

```

```
SET TALK OFF
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
SELECT derevo
LOCATE FOR bas_key=ALLTRIM(pkey)
IF FOUND()
CALCULATE MAX(VAL(key) ) TO s
```

```
APPEND BLANK
replace naimen WITH naimen_ ;
    , key WITH ALLTRIM(STR(s+1))+ "_ ";
    , pkey WITH bas_key;
    , summa_ with sum_
RETURN .t.
ELSE
RETURN .f.
ENDIF
```

```
PROCEDURE сумма_дочерних_узлов
lparameters bas_key
LOCAL s
SET TALK OFF
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
SELECT derevo
s=0
CALCULATE suM(summa_) FOR bas_key==ALLTRIM(pkey) to s
RETURN s
```

```
PROCEDURE сумма_дочерних_узлов_с_заменой
lparameters bas_key
LOCAL s
SET TALK OFF
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
SELECT derevo
*проверка на наличие дочерних узлов
IF получение_количества_дочерних_узлов(bas_key)=0
RETURN .f.
ENDIF
```

```
s=0
CALCULATE suM(summa_) FOR bas_key==ALLTRIM(pkey) to s
LOCATE FOR bas_key==ALLTRIM(key)
IF FOUND()
replace summa_ with s
RETURN .t.
ELSE
RETURN .f.
```

ENDIF

\* заменять сумму в узле суммой дочерних узлов

CLEAR

SET DEFAULT TO "u:\tree"

замена\_итог\_суммы("50\_")

PROCEDURE замена\_итог\_суммы

LPARAMETERS p,s

LOCAL s, k, x

\*p-ключ родительского элемента

? "вызов замена\_итог\_суммы ",p

IF NOT USED("derevo")

USE derevo IN 0

ENDIF

s=0

IF получение\_количества\_дочерних\_узлов(p)=0

? " узел ",p

s=получение\_суммы\_узла\_по\_ключу(p)

? s

ELSE

? "Просмотр всех дочерних узлов узла с кодом=",p

SCAN FOR ALLTRIM(pkey)==p

x=ALLTRIM(key)

k=RECNO()

? " pkey=",p," x=key= ",x

s=s+замена\_итог\_суммы(x)

?? "s=",s

GO k

ENDSCAN

LOCATE FOR p==ALLTRIM(key)

IF FOUND()

replace summa\_ WITH s

ENDIF

ENDIF

RETURN s

PROCEDURE получение\_количества\_дочерних\_узлов

LPARAMETERS lp\_key

IF NOT USED("derevo")

USE derevo IN 0

ENDIF

SELECT derevo

COUNT FOR lp\_key==ALLTRIM(pkey) TO b

RETURN b



```

PROCEDURE сумма_дочерних_узлов_с_заменой
lpparameters bas_key
LOCAL s
SET TALK OFF
IF NOT USED("derevo")
USE derevo IN 0
ENDIF
SELECT derevo
*проверка на наличие дочерних узлов
IF получение_количества_дочерних_узлов(bas_key)=0
RETURN .f.
ENDIF

s=0
CALCULATE suM(summa_) FOR bas_key==ALLTRIM(pkey) to s
LOCATE FOR bas_key==ALLTRIM(key)
IF FOUND()
replace summa_ with s
RETURN .t.
ELSE
RETURN .f.
ENDIF

```

```

PROCEDURE получение_суммы_узла_по_ключу
LPARAMETERS lp_key
IF NOT USED("derevo")
USE derevo IN 0
ENDIF

```

```

SELECT derevo
locate FOR lp_key==ALLTRIM(key)
IF FOUND()
RETURN summa_
ELSE
RETURN 0
ENDIF

```

Алгоритм решения на языке **VFP**: