17. ЕДНОСВЪРЗАН СПИСЪК

17.1. Общо описание

Едносвързаният списък е линейна структура от свързани еднотипни компоненти. Компонентите на едносвързания списък са динамични променливи от тип запис с две полета:

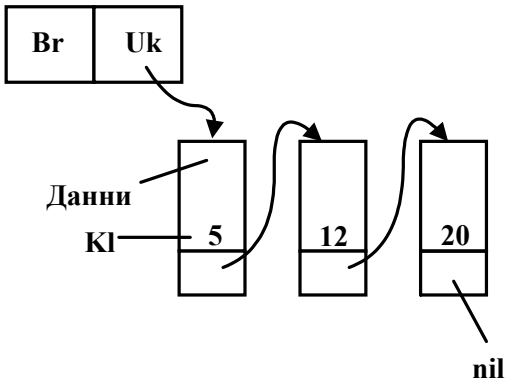
- информационно поле Inf - обикновено е от тип запис с полета, определени от конкретното предназначение на списъка. Тук ще приемем, че то има само две полета: Ime и Kl, където Kl е ключ, т.е. поле с уникална стойност за всяка компонента;
- указателно поле Uk - указва следващата компонента в едносвързания списък.

За да се организира едносвързан списък е нужна една променлива от тип указател, която да указва началото (първата компонента) на списъка. С цел да се създадат известни удобства обикновено се ползва и една целочислена променлива за брояч на компонентите в списъка. Поради това, едносвързаният списък ще присъства в програмата като променлива от тип запис с две полета: указател към началото на списъка Uk и брояч на компонентите Br (Фиг.17.1).

Списъкът може да бъде:

- неподреден - всяка следваща компонента се добавя на произволно място в списъка;
- подреден в нарастващ или намаляващ ред на стойностите на ключа;

Добавянето на компоненти в неподреден списък е по-лесно, защото всеки нов компонента се поставя в края на списъка. Подреденият списък е по-полезен - той може да се използва например за сортиране на масиви и файлове.



Фиг.17.1 Списък, подреден по ключ

Тук ще разгледаме основните операции със списък, подреден в нарастващ ред на стойностите на ключа. Те са:

1. Инициализация на едносвързан списък - привеждане на едносвързан списък в състояние “празен едносвързан списък”.
2. Добавяне на компонента.
3. Извличане на компонента.
4. Изтриване на компонента.

17.2. Дефиниране на тип едносвързан списък

```
Type
  TipKl=integer;     {Тип на ключа}
  TipInf=record      {Тип на информац. поле на комп. от списъка}
       Ime:string[25];
       Kl:TipKl
       end;
  TipUk=^TipEl;      {Тип на указател към комп. от списъка}
  TipEl=record       {Базов тип или тип на комп. от списъка}
       Inf:TipInf;   {Информац. поле на комп. от списъка}
       Uk:TipUk      {Указател към следв. комп. от списъка}
       end;
  TipStr=record      {Тип на структурата едносвързан списък}
       Uk:TipUk;     {Указател към началото на списъка}
       Br:integer    {Брой на компонентите в списъка}
       end;
```

17.3. Дефиниране на едносвързани списъци (променливи от тип едносвързан списък)

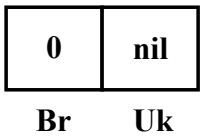
Едносвързаните списъци, нужни на програмата, се дефинират в раздела **Var**. Например с оператора

```
Var
  SpisA, SpisB, SpisX:TipStr;
```

са дефинирани три списъка с имена съответно SpisA, SpisB и SpisX.

17.4. Инициализиране на едносвързан списък

Под инициализиране на едносвързан списък се разбира обявяване на едносвързания списък за празен, а това се постига като се присвои нулева стойност на брояча на компонентите и стойност **nil** на указателя към началото на списъка (Фиг.17.2).



Фиг.17.2 Инициализиране на едносвързан списък

Едносвързани списъци могат да се инициализират чрез следната процедура, където Str е фиктивен параметър, който се замества с името на инициализирания едносвързан списък:

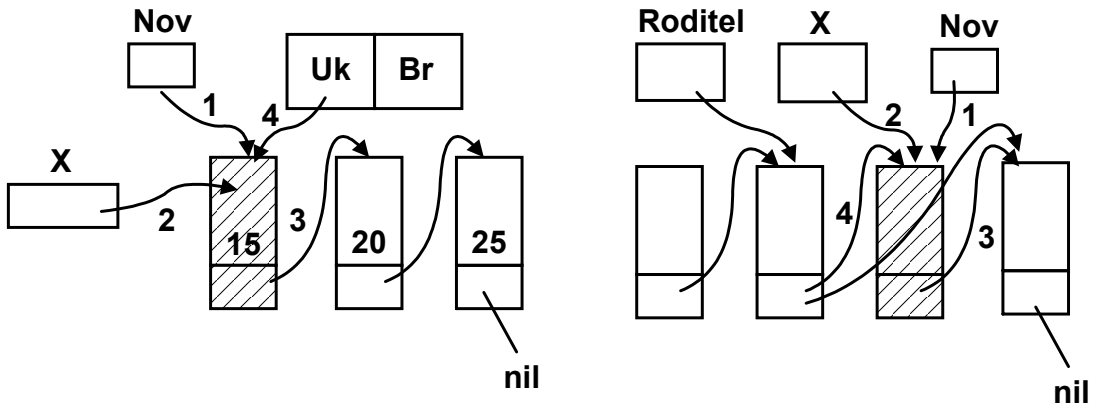
```
Procedure Init(Var Str:TipStr);
  Begin
    Str.Uk:=Nil; Str.Br:=0;
  End;
```

17.5. Добавяне на нова компонента към едносвързан списък

Както е показано на Фиг.17.3, за да се добави нова компонента към едносвързан списък трябва да се изпълнят посочените по-долу операции, като се използват допълнителните указатели Nov, указващ новосъздаваната компонента и Roditel, указващ родителя на новосъздаваната компонента.

1. Създаване на нова празна компонента - това става с помощта на процедурата New.

2. Записване в информационното поле на новата компонента съдържанието й.



Фиг.17.3 Добавяне на компонента

Следващите операции зависят от мястото, където попада новата компонента:

-Новата компонента попада в празен списък или в началото на непразен списък - трябва да се направи така, че:

3. Указателят на новата компонента да сочи следващата след него.

4. Указателят към списъка да сочи новата компонента, защото тя е начална.

-Новата компонента попада на друго място в списъка - трябва да се направи така, че:

3. Указателят на новата компонента да сочи следващата след него.

4. Указателят на компонентата-родител да сочи новата компонента.

Към едносвързан списък от декларирания тип може да се добави нова компонента например чрез функцията, дадена по-долу, където Str и X са фиктивни параметри. При обръщение към функцията фиктивните параметри се заместват съответно с името на едносвързан списък и променливата, чиято стойност трябва да запълни информационното поле на новата компонента.

```
Function DobPoKl(Var Str:TipStr;X:TipInf):boolean;
  Var
    Nov,              {Указател към новата компонента}
    Roditel:TipUk;    {Указател към родителя на новата компонента}
  Begin
    If MaxAvail>=SizeOf(TipEl)      {Има място за още компоненти}
       then begin
            New(Nov);
            Nov^.Inf:=X;
            If (Str.Uk=nil) or (Str.Uk^.Inf.Kl>=X.Kl)
            then begin       {Добавяне в празен списък или в началото на списъка}
              Nov^.Uk:=Str.Uk;
```

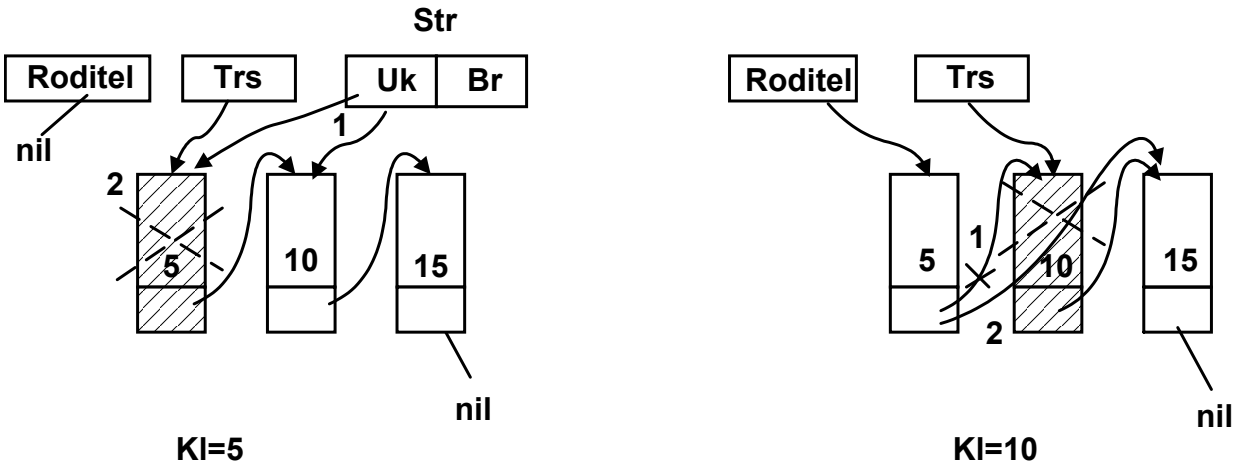
```
        Str.Uk:=Nov
      end
      else begin      {Търсене родител на новия елемент и добавяне}
      Roditel:=Str.Uk;
      While not((Roditel^.Uk=nil) or (Roditel^.Uk^.Inf.Kl>X.Kl)) do
        Roditel:=Roditel^.Uk;
      Nov^.Uk:=Roditel^.Uk;
      Roditel^.Uk:=Nov
      end;
      Str.Br:=Str.Br+1;
      DobPoKl:=true
    end
  else DobPoKl:=false
End;
```

Тази функция придобива стойност true, когато се справи с добавянето на нова компонента (в ОП има място за нова компонента), и стойност false, когато добавянето е невъзможно, поради това, че в ОП няма място за нова компонента.

17.6. Изтриване на компонента със зададена стойност на ключовото поле

За да се изтрие компонента от едносвързан списък, трябва най-напред да се намерят адресите на компонентата, съдържаща зададения ключ, и на нейния родител и след това да се изпълни поредицата операции, показана на Фиг.17.4.

От едносвързан списък може да се изтрие компонента например чрез функцията, дадена по-долу, където Str и Kl са фиктивни параметри. При извикването й те се заместват съответно с името на едносвързания списък и стойността на ключовото поле на компонентата, която трябва да се изтрие.



Фиг.17.4 Изтриване на компонента

```
Function IztrPoKl(Var Str:TipStr;Kl:TipKl):boolean;
  Var
    Trs,              {Указател към търсената компонента}
    Roditel:TipUk;    {Указател към родителя на търсената компонента}
  Begin
```

```
  If Str.Uk=nil
  then IztrPoKl:=false     {Списъкът е празен}
  else begin               {Търсене на компонента за изтриване}
    Trs:=Str.Uk; Roditel:=nil;
    While not((Trs^.Uk=nil) or (Trs^.Inf.Kl=Kl)) do
      begin
        Roditel:=Trs;
        Trs:=Trs^.Uk
      end;
      If Trs^.Inf.Kl<>Kl
      then IztrPoKl:=false  {Не е намерена комп. със задад. ключ}
      else begin
        If Roditel=nil
        then         {Първата комп. ли е за изтриване?}
            Str.Uk:=Str.Uk^.Uk              {Изтрива се първата}
        else Roditel^.Uk:=Trs^.Uk;          {Изтрива се непърва}
      Dispose(Trs);
      Str.Br:=Str.Br-1;
      IztrPoKl:=true
    end
  end;
End;
```

17.7. Извличане на компонента със зададена стойност на ключовото поле

От едносвързан списък от декларирания тип може да се извлече компонента например чрез функцията, дадена по-долу, където Str и Kl и X са фиктивни параметри. При обръщение към функцията те се заместват съответно с името на едносвързания списък, стойността на ключовото поле на компонентата, чието информационно поле трябва да се извлече и променливата, която извлича информационното поле.

```
Function IzvlPoKl(Str:TipStr;Kl:TipKl;Var X:TipInf):boolean;
  Var
    Trs,              {Указател към търсената компонента}
    Roditel:TipUk;    {Указател към родителя на търсената компонента}
  Begin
    If Str.Uk=nil
    then IzvlPoKl:=false     {Списъкът е празен}
    else begin               {Търсене на компонента за извличане}
      Trs:=Str.Uk; Roditel:=nil;
      While not((Trs^.Uk=nil) or (Trs^.Inf.Kl=Kl)) do
      begin
        Roditel:=Trs;
        Trs:=Trs^.Uk
      end;
      If Trs^.Inf.Kl<>Kl
      then IzvlPoKl:=false  {Не е намерена комп. със задад. ключ}
      else begin                    {Намерена е компонента със задад. ключ}
        If Roditel=nil              {Първата комп. ли е за извличане?}
        then X:=Str.Uk^.Inf         {Извлича се първата}
```

```
      else X:=Roditel^.Uk^.Inf;     {Извлича се непърва}
      IzvlPoKl:=true
    end
  end
End;
```

Функцията IzvlPoKl връща стойност true, когато е извлечено информационното поле на компонентата с посочен ключ, и стойност false, когато в списъка не е намерена компонента с посочената ключова стойност.