

18. ДВУСВЪРЗАН СПИСЪК

18.1. Общо описание

Двусвързаният списък представлява линейна структура от свързани еднотипни компоненти. Компонентите на двусвързания списък са динамични променливи от тип запис с три 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;

- две указателни полета *Lv* и *Ds* - указващи предходната и следващата компоненти в двусвързан списък.

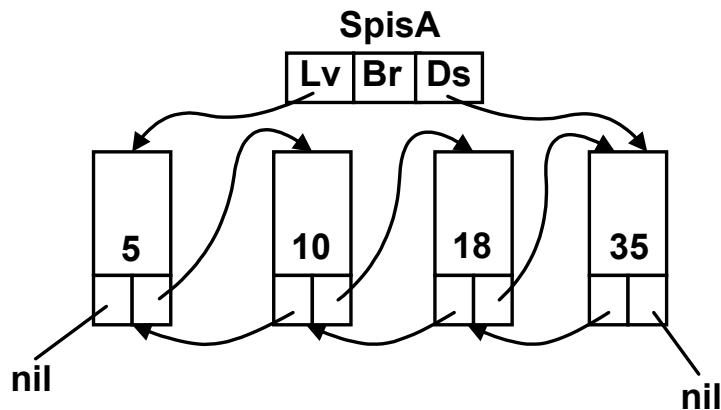
За да се организира двусвързан списък е нужна една променлива от тип указател, която да указва началото (първата компонента) на списъка. С цел да се създадат известни удобства обикновено се ползва и една целочислена променлива за брояч на компонентите в списъка. Поради това, двусвързаният списък ще присъства в програмата като променлива от тип запис с две полета: указател към началото на списъка *Uk* и брояч на компонентите *Br* (Фиг.18.1).

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 елемент се поставя в края на списъка. Подреденият списък обаче е по-полезен - той може да се използва например за сортиране на едномерни масиви и файлове.



Фиг.18.1 Списък, подреден по ключ

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

1. Инициализация на двусвързан списък - привеждане на двусвързан списък в състояние "празен двусвързан списък".
2. Добавяне на компонент.
3. Извличане на компонент.

4. Изтриване на компонент.

18.2. Дефиниране на тип двусвързан списък

Type

```
TipKl=integer;    {Тип на ключа}
TipInf=record     {Тип на информац. поле на комп. от списъка}
  lme: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}
  Lv,            {Указател към лявата съседна компонента}
  Ds:TipUk      {Указател към дясната съседна компонента}
end;
TipStr=record     {Тип на структурата двусвързан списък}
  Lv,            {Указател към левия край на списъка}
  Ds:TipUk;     {Указател към десния край на списъка}
  Br:integer     {Брой на компонентите в списъка}
end;
```

18.3. Дефиниране на двусвързани списъци (променливи от тип двусвързан списък)

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

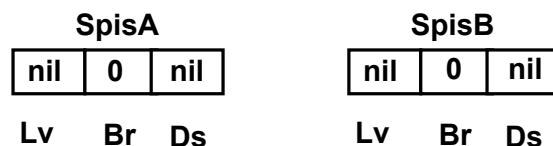
Var

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

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

18.4. 18.4. Инициализиране на двусвързан списък

Под инициализиране на двусвързан списък се разбира обявяване на двусвързания списък за празен, а това се постига като се присвои нулева стойност на брояча на компонентите и стойност **nil** на указателите към левия и десния край на списъка (Фиг.18.2).



Фиг.18.2 Инициализирани списъци

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

```
Procedure Init(Var Str:TipStr);
Begin
```

```

Str.Lv:=Nil; Str.Ds:=Nil; Str.Br:=0;
End;

```

18.5. Добавяне на нова компонента към двусвързан списък

Към двусвързан списък от декларирания тип може да се добави нова компонента например чрез функцията, дадена по-долу. Str и X са фиктивни параметри, които се заместват съответно с името на двусвързан списък и променливата, чиято стойност става стойност на информационното поле на новата компонента.

```

Function DobPoKl(Var Str:TipStr;X:TipInf):boolean;
Var
  Nov,                               {Указател към новата компонента}
  SledNov:TipUk;                     {Указател към комп., следваща новата компонента}
Begin
  If MaxAvail>=SizeOf(TipEl)
  then begin
    Str.Br:=Str.Br+1; DobPoKl:=true;
    New(Nov); Nov^.Inf:=X;
    If Str.Lv=nil
    then begin {Добавяне в празен списък}
      Nov^.Ds:=nil;
      Nov^.Lv:=nil;
      Str.Lv:=Nov;
      Str.Ds:=Nov;;
      Exit
    end;
    If X.Kl>Str.Ds^.Inf.Kl
    then begin {Добавяне след последната компонента}
      Nov^.Ds:=Nil;
      Nov^.Lv:=Str.Ds;
      Str.Ds^.Ds:=Nov;
      Str.Ds:=Nov;
      Exit
    end;
    SledNov:=Str.Lv; {Търсене място за добавяната компонента}
    While SledNov^.Inf.Kl<X.Kl do
      SledNov:=SledNov^.Ds;
    If SledNov^.Lv=nil
    then begin {Добавяне пред първата компонента}
      Nov^.Lv:=Nil;
      Nov^.Ds:=Str.Lv;
      Str.Lv^.Lv:=Nov;
      Str.Lv:=Nov
    end
    else begin {Добавяне пред непърва компонента}
      Nov^.Lv:=SledNov^.Lv;
      Nov^.Ds:=SledNov^.Lv^.Ds;
      SledNov^.Lv^.Ds:=Nov;
      SledNov^.Lv:=Nov
    end;
  end
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. Тя намира мястото на новата компонента в списъка и я добавя.

18.6. Търсене на указател към компонента, съдържаща зададен ключ

Търсенето на указател към компонента от двусвързан списък, съдържаща зададена ключова стойност, е задача на следващата подпрограма. Тя връща намерения указател или **nil**, ако в списъка няма компонента с посочената ключова стойност.

```

Function TrsUk(Str:TipStr;Kl:TipKl):TipUk;
Var
  Trs:TipUk;      {Указател към търсената компонента}
Begin
  If Str.Lv=nil
  then TrsUk:=nil {Списъкът е празен, компонента не е намерена}
  else begin      {Следва търсене на компонентата със задад. ключ}
    Trs:=Str.Lv;
    While not((Trs^.Ds=nil) or (Trs^.Inf.Kl=Kl)) do
      Trs:=Trs^.Ds;
    If Trs^.Inf.Kl=Kl
    then TrsUk:=Trs {Намерена е компонента}
    else TrsUk:=nil; {Не е намерена компонента}
    end
  End;

```

18.7. Извличане на компонента по зададена стойност на ключовото поле

От двусвързан списък може да се извлече компонента например чрез функцията IzvlPoKl, дадена по-долу. Фиктивните параметри Str, Kl и X се заместват съответно с името на двусвързания списък, стойността на ключовото поле на компонентата, чието информационно поле трябва да се извлече, и променливата, която извлича информационното поле. Функцията IzvlPoKl използва функцията TrsUk, дадена по-горе, за да намери адреса на компонентата, подлежаща на извличане.

```

Function IzvlPoKl(Str:TipStr;Kl:TipKl;Var X:TipInf):boolean;
Var
  Trs:TipUk;      {Указател към компонентата със задад. ключ}
Begin
  Trs:=TrsUk(Str,Kl);
  If Trs<>nil      {Намерена е компонента със зададения ключ}
  then begin
    X:=Trs^.Inf;
    IzvlPoKl:=true
  end

```

```

else IzvIPoKI:=false
End;

```

Функцията IzvIPoKI 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.

18.8. Изтриване на компонента със зададена стойност на ключовото поле

От двусвързан списък от декларирания тип може да се изтрие компонента например чрез функцията, дадена по-долу, където Str и Kl са фиктивни параметри. При извикване на функцията те се заместват съответно с името на двусвързания списък и стойността на ключовото поле на компонентата, която трябва да се изтрие.

```

Function IztrPoKI(Var Str:TipStr;Kl:TipKl):boolean;
Var
  Iztr:TipUk;      {Указател към комп., която трябва да се изтрие}
Begin
  Iztr:=TrsUk(Str,Kl);
  If Iztr<>nil      {Намерена ли е компонента, съдържаща зададения ключ?}
  then begin      {Намерен е}
    If(Iztr^.Lv=Nil) and (Iztr^.Ds=Nil)
    then begin    {Изтриваната е единствена компонента в списъка}
      Str.Lv:=Nil;Str.Ds:=Nil
    end
    else if Iztr^.Lv=Nil
    then begin    {Изтриваната е първа компонента в списъка}
      Str.Lv:=Str.Lv^.Ds;
      Str.Lv^.Lv:=Nil
    end
    else if Iztr^.Ds=Nil
    then begin    {Изтриваната е последна компонента}
      Str.Ds:=Str.Ds^.Lv;
      Str.Ds^.Ds:=Nil
    end
    else begin    {Изтриваната е междинна компонента}
      Iztr^.Lv^.Ds:=Iztr^.Ds;
      Iztr^.Ds^.Lv:=Iztr^.Lv
    end;
    Dispose(Iztr);
    Str.Br:=Str.Br-1;
    IztrPoKI:=true
  end
  else IztrPoKI:=false
End;

```

Функцията IztrPoKI изтрива компоненти от двусвързан списък на две стъпки:

1. Намира адреса на компонентата, която трябва да се изтрие. За тази цел тя използва функцията TrsUk, която разгледахме по-горе.
2. Изтрива компонентата с намерения адрес.

Функцията `IztrPoKI` връща стойност `true`, когато е изтрила компонентата с посочената ключова стойност, и стойност `false`, когато не е намерила компонентата с посочената ключова стойност.