14. УКАЗАТЕЛИ. ДИНАМИЧНИ ПРОМЕНЛИВИ. ДИНАМИЧНИ СТРУКТУРИ

14.1. Указатели и динамични променливи

От изложеното до тук е ясно, че в програмите на Паскал можем да декларираме константи и променливи от стандартен или дефиниран от потребителя тип. При това изяснихме, че за константите и променливите, дефинирани в главната програма, се заделя място в областта от ОП за статични обекти и това място се пази докато програмата се изпълнява. За константите и променливите, дефинирани в подпрограма, се заделя място в системния стек, когато подпрограмата се активира, и това място се пази докато тя е активна.

В програмите на Паскал обаче могат да се използват и такива променливи, които се създават от програмата в онзи момент от нейното изпълнение, в който те са нужни, и се унищожават, когато повече не са нужни. Освободеното по този начин място от ОП може да се използва за други такива променливи. Този вид променливи се наричат *динамични променливи* и те са свързани тясно с така наречените променливи от тип указател или накратко *указатели*. Връзката се изразява в следното:

- указателят е променлива или константа, която има за стойност адреса на някоя динамична променлива или стойността nil. Указател, на който е присвоена стойност nil, не указва никаква динамична променлива;
- името на динамичната променлива се получава от името на указателя, който я указва, като последното се допълни със знака **^**, т.е., ако името на указателя е P, то името на указваната от него динамичната променлива е P^;

14.1.1. Дефиниране на тип указател

Както е известно, между типа на дадена променлива и типа на нейните стойности трябва да има съответствие. Например променлива от целочислен тип може да има само целочислена стойност, променлива от реален тип може да има целочислени и нецелочислени стойности и т.н. Стойности на указателя могат да бъдат само адреси на клетки от ОП. Но типът на указателите не се свързва със стойностите на указателите, а с стойностите на указваните динамични променливи, който се нарича *базов тип*. Или типове указатели се дефинират както следва:

Type

Име на типа указатели = **^***Име на базов тип;*

Както се вижда от този пример, при дефиниране на тип указатели не се използва запазена дума (**array**, **set of**, **record** или **file of**), а специалният знак "**^**", който съответства на думата "указващ".

Когато базовият тип (типът на указваните динамични променливи) е стандартен, той само се посочва, например

```
Type
  TipUkInt = ^ integer;
  TipUkRl = ^ real;
  TipUkCh = ^ char;
  TipUkStr = ^ string;
```

Когато базовият тип не е стандартен и не съдържа поле от дефинирания указателен тип, той може да се дефинира както преди дефиниране на типа, така и

след това, т.е. допустим е всеки от следните два варианта:

```
Type
  TipZap=record
           Ime:string[25];
           EGN:string[10]
         end;
  TipUkZap=^TipZap;
```

или

```
Type
  TipUkZap=^TipZap;
  TipZap=record
           Ime:string[25];
           EGN:string[10]
         end;
```

В Турбо Паскал съществува и стандартен тип указател, наречен ***pointer***. На указател от тип ***pointer*** могат да се присвояват стойностите на указателите от всякакъв базов тип (адресите на динамични променливи от всякакъв тип). Съществуват и други удобства, предлагани от стандартния тип указател ***pointer***.

14.1.2. Дефиниране на указатели

Указатели от съответни типове се дефинират както следва:

```
UkInt  : TipUkInt;
UkRl   : TipUkRl
UkCh   : TipUkCh;
UkStr  : TipUkStr;
UkZap  : TipUkZap;
Uk     : pointer;
```

Тези указатели, указваните от тях динамични променливи и типовете на динамичните променливи са дадени в следната таблица:

Указател	Динамична променлива	Тип на указваната динамична променлива
UkInt	UkInt^	Integer
UkRl	UkRl^	Real
UkCh	UkCh^	Char
UkStr	UkStr^	String
UkZap	UkZap^	TipZap
Uk	Uk^	Всеки допустим базов тип

14.1.3. Създаване на динамични променливи

Динамични променливи се създават посредством процедурата **New**. Обръщението към нея има вида:

New (*Име на указател*);

Например, можем да създадем динамична променлива UkInt^, указвана от указателя UkInt, чрез оператора-обръщение New (UkInt).

Изпълнението на процедурата New води до:

- създаване на динамичната променлива, т.е. резервиране на памет за нея;
- присвояване адреса на създадената динамична променлива на указателя, който е посочен като действителен параметър на процедурата New.

За проверка дали има достатъчно свободна памет може да се използва функцията **MaxAvail**, която връща като резултат размера на най-голямата незаета част от ОП.

14.1.4. Унищожаване на динамични променливи

Динамични променливи се унищтожават посредством процедурата **Dispose**. Обръщението към нея има вида:

Dispose(*Име на указател*);

Например, динамичната променлива UkInt**^**, указвана от указателя UkInt, можем да унищожим чрез оператор-обръщение Dispose (UkInt). В резултат на изпълнението на процедурата Dispose от ОП изчезва динамичната променлива UkInt^ и се освобождава мястото от паметта, заемано от нея, но на указателя UkInt не се присвоява стойност **nil**.

14.1.5. Операции с указатели

Допустими са само операциите присвояване и сравняване на указатели и то при съблюдаване на следните ограничения:

а) За операцията присвояване:

- на всеки указател, независимо от неговия тип, може да се присвои стойност **nil**;
- на един указател може да се присвои стойността на друг указател, само ако са от един и същ тип, т.е. операцията p2:=p1 с указателите p1 и p2 е допустима, само ако те са еднотипни.

б) За операцията сравняване:

- разрешени са само операциите p1=p2 и p1<>p2 и то ако p1 и p2 са от един и същи тип;
- всеки указател, независимо от неговия тип, може да се сравни със стойността **nil** (p1=**nil** p1<>**nil**).

14.1.6. Операции с динамични променливи

По-горе беше посочено, че типът на динамичните променливи може да бъде някой от стандартните типове или дефиниран нестандартен тип. Операциите, допустими за дадена динамична променлива, са тези, които са допустими за нейния тип (посочения базов тип при дефиниране типа на нейния указател). Когато базовият тип е запис, достъпът до полетата е обичайния за полета на записи. Например, ако UkZap е указател от базов тип TipZap,то UkZap^ е динамичната променлива, указвана от UkZap, а UkZap^.Ime, UkZap^.EGN са две нейни полета.

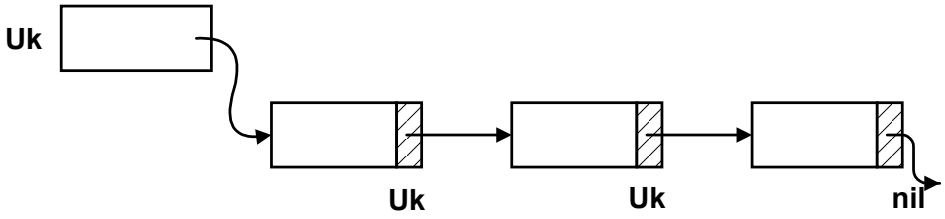
14.2. Свързани динамични променливи. Динамични структури

Базовият тип може да съдържа полета от указателен тип. В този случай обаче той трябва да се дефинира след дефиниране указателния тип. Ето един такъв пример:

```
Type
  TipUkZap=^TipZap;       {Указателен тип}
  TipZap=record   {Базов тип}
            Ime:string[25];
            EGN:string[10];
```

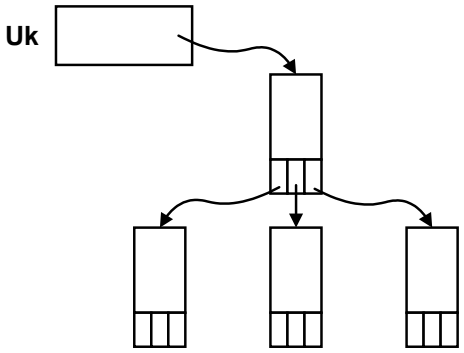
```
    Uk:TipUkZap     {Полето Uk от указателния тип TipUkZap}
end;
```

Възможността базовият тип да съдържа полета от указателен тип има голямо практическо значение. Можем да дефинираме нова динамична променлива, указвана от указателното поле на вече създадена динамична променлива. Можем да дефинираме следваща динамична променлива, указвана от указателното поле на току що създадена динамична променлива и т.н. Свързаните по този начин динамични променливи образуват структура от данни, наречена линейна динамична структура (Фиг.14.1).



Фиг.14.1 Линейна динамична структура

Динамичната променлива може да има няколко указателни полета, например 3. Това означава, че можем да създадем 3 нови динамични променливи, указвани от тези указателни полета. По-нататък можем да създадем нови 9 динамични променливи, указвани от указателните полета на трите динамични променливи и т. н. Свързаните по този начин динамични променливи образуват структура от данни, наречена разклонена динамична структура (Фиг.14.2).



Фиг.14.2 Разклонена динамична структура

Отделните динамични променливи от дадена динамична структурата се наричат елементи, компоненти или възли на структурата.