Тема 8

Структурирани типове данни в Pascal. Масиви, записи.

1. Масиви

Много често в математиката, икономиката, физиката и др. научни област се налага да се обработват съвкупности от данни с еднакъв смисъл и от един и същи тип. Напр.: температурата на въздуха и водата за всеки ден от месеца, измерена в определен час от денонощието; месечните заплати на служителите на една фирма, приходите на фирмата за всеки месец от годината, валутния курс за всеки ден. За по-удобната обработка на данните в такива случаи е удобно те да се подредят, номерират и да образуват нова стуктура от данни, наречена масив. Отделните компоненти на масива се наричат елементи елементи на масива.

Масивът е крайна, наредена съвкупност от еднотипни данни.

Всеки масив се характеризира с име, брой елементи и тип на елементите. Елементите на масива могат да бъдат от всеки допустим в езика тип, включително и структуриран. Типът на елементите на масива се нарича базов тип на масива.

Елементите на масива се означават с индекс, който показва позицията му в масива и по тази причина елементите на масива се наричат още индексирани променливи. Боят на елементите на масива се фиксира при декларирането на масива и не се променя в програмата. Индексите на елементите на масива могат да бъдат константи, променливи или изрази от всеки дискретен тип.

Всички масиви трябва да се декларират в декларативната част на програмата в частта за променливи **VAR**. Изпозват се служебните думи **ARRAY…OF** които значат **масив ... от .**

име: **ARRAY** [***диапазон***] **OF** ***тип***;

където:

име – име на масива;
диапазон определя началната и крайната стойност за индексите на елементите на масива; началната и крайната стойност се разделят с две точки.
тип - определя типа на елементите на масива.

Пример:
VAR
C: ARRAY[1..6] of real;

Когато в програмата се използват повече от един масива от един и същи тип (т.е. с еднакъв брой и тип на елементите) се препоръчва този ти да се декларира в раздела **TYPE** на декларативната част на програмата. Форматът за деклариране в този случай е следният:

TYPE ***име на типа*** = **ARRAY**[***тип на индекса***] **OF** ***тип на елементите***;

В областта за деклариране на променливи в този случай трябва да се включи деклариране на променливите от този тип масив по следния начин
VAR ***списък от имена на масиви : име на тип;***
Пример:

TYPE masiv = **ARRAY**[1..12] **OF** REAL;

VAR
B, C : masiv***;***

Според начина на подреждане на елементите си, респективно според броя на индексите на елементите, масивите са едномерни, двумерни, тримерни и т. н.

Елементите на един масив могат да бъдат и от структуриран тип. Когато в качеството на базов тип за масива се използва друг масив се поучава "масив от масиви", който се нарича още многомерен масив.

При работа с масиви е възможно да се обработи целия масив или да се обработи масива по елементи (да се обработят всички елементи поотделно).

Единствената допустима обработка на цял масив е приложима за два еднотипни масива, когато всички стойности на елементите на един масив се присвояват на съответните елементи на друг масив.

```
TYPE masiv1=ARRAY[1..15] of INTEGER;
VAR
      A, B : masiv1;
……
…….
A:=B;
```

Размерността на масивите се ограничава единствено от наличната свободна оперативна памет (ОП) за разполагане на стойностите на елементите. Стойностите на елементи се разполагат в ОП последователно в зависимост от индексите си.

2. Едномерни масиви

Едномерният масив отговаря на математическото понятие вектор.
Елементите на едномерния масив са с един индекс.

C[1]	C[2]	C[3]	C[4]	C[5]	C[6]

Променлива едномерен масив се дефинира по следния начин

име: **ARRAY** [***диапазон***] **OF** ***тип***;

където:

име – име на масива;
диапазон определя началната и крайната стойност за индексите на елементите на масива; началната и крайната стойност се разделят с две точки (..).
тип - определя типа на елементите на масива.

Пример:

T: ARRAY[1..16] of real;

Вътре в тялото на програмата се работи с елементите на масива по същия начин както с променливи от съответния тип.

Обръщението към отделните елементи става по следния начин:

***име*[*индекс*]**

където ***име*** е името на променливата масив, а ***индекс*** е израз със стойност в диапазона на индексите.

Пример: C[5]

Обработването на масивите е по елементи. За да се обработят всички елементи на един масив най-подходящо е операторът (операторите) за съответната обработка да се включи в цикъл **FOR,** в който управляващата променлива на цикъла е и в ролята на текущ индекс за елементите на масива.

Примери за обработка на едномерен масив:

Въвеждане на стойности за елементите на масив C с 10 елемента
FOR I:=1 TO 10 DO READ(C[I]);

Извеждане на стойностите на елементите на C с 10 елемента
FOR I:=1 TO 10 DO WRITELN(C[I]);

Нулиране на стойностите на елементите на масив C с 10 елемента
FOR I:=1 TO 10 DO C[I]:=0;

Пример: Да се намери средната стойност на елементите на едномерен масив T, който се състои от 6 реални елемента.

```
PROGRAM SRST;
VAR
I:INTEGER;
S,SR:REAL;
(*S-SUMA, SR-SREDNA STOINOST*)
T:ARRAY [1..6] OF REAL;
BEGIN
WRITELN('VAVEDETE ELEMENT NA MASIVA');
FOR I:=1 TO  6 DO
    BEGIN
    WRITE ('T[',I,']=');
    READLN(T[I])
    END;
S:=0.0;
FOR I:=1 TO 6 DO
    S:=S+T[I];
SR:=S/6;
WRITELN ('SREDNA STOINOST SR=', SR)
END.
```

Пример: Да се намери сумата на положителните елементи на масив T[6], елементите на който са реални стойности.

```
PROGRAM SUMP;
{сума на положителните елементи}
VAR
I:INTEGER;
S:REAL;
T:ARRAY [1..6] OF REAL;
BEGIN
WRITELN('VAVEDETE ELEMENT NA MASIVA');
FOR I:=1 TO  6 DO
    BEGIN
    WRITE ('T[',I,']=');
    READLN(T[I])
    END;
S:=0.0;
FOR I:=1 TO 6 DO
    IF T[I]>0 THEN S:=S+T[I];
WRITELN ('SUMATA NA POLOZITELNITE ELEMENTI E ', S)
END.
```

Условие на задачата:

Да се въведат произволни реални стойности за елементите на масив T[6] .Да се формира нов едномерен масив U който да съдържа само тези стойности на масив T[6],който са по-големи от 0 и по-малки от 10.

Разяснения:

Входни данни:

T[6] *- Входен масив*

Междинни данни:

i *- управляваща променлива на цикъла и индекс на текущ елемент на масива.*

j *- текущ индекс за ново формираният масив.*

Изходнни данни:

U[6] *- ново формиран масив*

```
program Newarr ;
const N=6;
type vector=Array [1..n] of real;
Var I,J:integer;
T,U:vector;
Begin
for I:=1 to N do
Begin
Write('Елемент',I,':');
ReadLn (T[I])
End;
J:=0;
for I:=1 to N do
If (T[I]>0) and (T[I]<10) then
Begin
J:=J+1;
U[J]:=T[I]
End;
For I:= 1 to J do
Write ('U[',I,']=',U[I]:10:2)
End.
```

3. Двумерни масиви

В двумерния масив елементите са подредени по редове и стълбове. Той отговаря на математическото понятие матрица или таблица. Елементите на двумерния масив са с два индекса: първият определя номера на реда, а вторият – номера на стълба за съответния елемент.

	J=1	J=2	J=3	J=4	J=5
I=1	B[1,1]	B[1,2]	B[1,3]	B[1,4]	B[1,5]
I=2	B[2,1]	B[2,2]	B[2,3]	B[2,4]	B[2,5]
I=3	B[3,1]	B[3,2]	B[3,3]	B[3,4]	B[3,5]
I=4	B[4,1]	B[4,2]	B[4,3]	B[4,4]	B[4,5]

B[I,J] където I=1, 2, … , 4 (индекс по ред)
J=1, 2, … , 5 (индекс по стълб)

Променлива двумерен масив се декларира по следния начин:

име: **ARRAY** [***диапазон1, диапазон2***] **OF** ***тип***;

където
диапазон1 определя началната и крайната стойност за индексите по ред на елементите на масива;
диапазон2 - определя началната и крайната стойност за индексите по стълб на елементите на масива.

Пример:
TT: ARRAY[1..5, 1..8] OF REAL;

Обръщането към елемент на двумерен масив става чрез два индекса:

***име*[*индекс1, индекс2*]**

където ***индекс1*** и ***индекс2*** са целочислени изрази със стойности съответно в ***диапазон1*** и ***диапазон2***.

Пример:

B[3,5]

Обработването на масивите е по елементи. За да се обработят всички елементи на един двумерен масив е най-подходящо съответния оператор да се включи в структура на два вложени цикъла **FOR**. Ако управляващата променлива на външния цикъл е в ролята и на текущ индекс по ред за елементите на масива, а управляващата променлива на вътрешния цикъл е и в ролята на текущ индекс по стълб за елементите на масива, масивът се обработва по редове.

За да се обработи масива по стълбове във външния цикъл се изменя втория индекс (по стълб), а във вътрешния – първия индекс (по ред).

Примери за обработка на двумерен масив:

Въвеждане на стойности за елементите на масив C с 5 реда и 8 стълба. Въвеждането е по редове.

```
FOR I:=1 TO 5 DO
FOR J:=1 TO 8 DO
READLN(C[I,J]);
```

Въвеждане на стойности за елементите на масив C с 5 реда и 8 стълба. Въвеждането е по стълбове.

```
FOR J:=1 TO 8 DO
FOR I:=1 TO 5 DO
READLN(C[I,J]);
```

Извеждане на стойностите на елементите на С с 6 реда и 6 стълба. Извеждането е по редове.

```
FOR I:=1 TO 6 DO
FOR J:=1 TO 6 DO
WRITELN(C[I,J]);
```

Нулиране на стойностите на елементите на масив С с 9 реда и 7 стълба. Обработването е по редове.

```
FOR I:=1 TO 9 DO
FOR J:=1 TO 7 DO
C[I,J]:=0;
```

Пример: Да се състави програма за сумиране на елементите на двумерен масив Т, с 6 реда и 5 стълба и елементите му са от реален тип.

```
PROGRAM SUM;
VAR
    I, J: INTEGER;
    SUMA: REAL;
    T: ARRAY[1..6, 1..5] OF REAL;
BEGIN
WRITELN;
WRITELN('VAVEDETE ELEMENTITE NA MASIVA');
WTITELN;
FOR I:=1 TO 6 DO
FOR J:= 1 TO 5 DO
    BEGIN
    WRITE('T[',I,',',J,']=');
    READLN(T[I,J])
    END;
SUMA:=0.0;
FOR I:=1 TO 6 DO
FOR J:= 1 TO 5 DO
    SUMA:=SUMA + T[I,J];
WRITELN;
WRITELN('SUMATA NA ELEMENTITE NA MASIVA E ', SUMA:9:3)
END.
```

Пример: Да се състави програма за нулиране на отрицателните елементи на двумерен масив

```
PROGRAM NULIRANE;
{nulirane na otricatelnite elementi na masiv}
VAR
    I, J, N : INTEGER;
    P : REAL;
    A : ARRAY[1..50,1..50] OF REAL;
BEGIN
WRITELN;
{vavezdane broj na redovete i stylbovete na masiva}
REPEAT
    WRITE('VAVEDI N<=50 ');
    READ(N)
UNTIL (N>=1) AND (N<=50);
{vavezdane na stojnosti na elementite na masiva}
FOR I:=1 TO N DO
FOR J:=1 TO N DO
  BEGIN
  WRITE('VAVEDI A[',I,',',J,'] ');
  READ(A[I,J])
  END;
{nulirane na otricatelnite elementi na masiva}
FOR I:=1 TO N DO
FOR J:=1 TO N DO
    IF A[I,J]<0 THEN A[I,J]:=0;
WRITELN;
{izvezdane na prerabotenia masiv}
FOR I:=1 TO N DO
FOR J:=1 TO N DO
  WRITELN('A[',I,',',J,']=',A[I,J]:5:2);
END.
```

4. Записи

Масивът е структуриран тип данни, който се състои от еднотипни елементи. В много практически случаи се налага да се обединят в обща логическа сруктура данни за различни характеристики на един обект, които са от различен тип. Напр.: информационната сустема на ТУ-Габрово съдържа следните данни за всеки студент: име, фамилия, ЕГН, факултетен номер, специалност, факултет, курс, група, успех по дисциплини, среден успех, платил/не платил такса и др. За решаване на задачи от подобен тип в Pascal е включен структуриран тип данни, наречен запис. Записът е структуриран тип данни, който се състои от предварително определ брой елементи от различен тип. Елементите на записа се наричат полета. Казано с други думи, записът се състои от полета от различен тип.

Задължително е в декларативната част да се декларира (дефинира) типа запис. Общият формат на дефинирането на тип запис е следния:

```
TYPE
    име на тип = RECORD
                 име на полето:тип на полето;
                 име на полето:тип на полето;
                 ...
```

***име на полето:тип на полето*;**
END;

VAR
***име на променлива от тип запис: име на типа*;**

всяко от полетата на записа се означава с име и се разглежда като променлива от дадения тип. Препоръчва се името на полето да е уникално не само за записа, но и в рамките на цялата програма.

Действия с цели записи

Единствената допустима операция с цели записи е присвояването на една променлива от тип запис на друга променлива от тип запис. В този случай е задължително двете променливи да са еднотипни, т.е. броят и типа на полетата им да съвпадат.

P:=Q;

Този запис означава, че на полетата на променливата-запис P ще се присвоят стойностите на съответните полета от променливата-запис Q.

Действия с отделни полета на запис

ПО-често се налага обработването на отделните полета на записа. В този случай полето се използва по същия начин като проста променлива от съответния тип. Достъпът (обръщането) към поле на запис става чрез т.нар. **съставно име**. То се състои от името на записа и името на полето, разделени с точка. В общ вид това се записва по следния начин:

име на променлива тип запис . име на поле

Напр.: P.FN означава поле FN от запис P

Присвояване на стойност на поле от запис

P.FN:=107287;

Въвеждане и извеждане на стойности на поле от запис

READLN(P.FN**);**
WRITELN(P.FN)

Включване на поле от запис в израз

A:=P.FN*2;

Оператор за колективна обработка на полета от запис

Както се вижда от посочените примери, обтъщането към поле от запис става чрез дълго име. В случаите когато ще се обработват всички полета от записа е особено подходящо да се използва операторът за колектовна обработка на полетата от запис – **WITH.** Общият вид на този оператор е от вида:

WITH *променлива тип запис* DO *оператор*;

В този случай, след като веднъж сме указали името на записа с който се работи чрез оператор **WITH**, може да се използва само името на полето, а не съставното име. Ако ще обработваме всички полета на записа, то се налага операторът след **DO** да е съставен оператор.

Пример: Да се въведат данни за N на брой студенти, които да включват информация за името и средния успех. Да се намери броя на студентите с отличен успех (>5.50) и да се отпечатат имената им.

PROGRAM ZAPISI;

```
TYPE  STUDENT=RECORD
                 IME:STRING[30];
                USPEH:REAL
                END;
VAR
ST:ARRAY[1..50] OF STUDENT;
N, I, BR: INTEGER;
BEGIN
WRITELN;
REPEAT
 WRITE('VAVEDETE BROJ STUDENTI N=');
 READLN(N)
UNTIL (N>1) AND (N<=50);
WRITELN('VAVEDETE DANNI ZA STUDENTITE:');
FOR I:=1 TO N DO
 BEGIN
 WRITE('IME:'); READLN(ST[I].IME);
 WRITE('SREDEN USPEH:'); READLN(ST[I].USPEH)
 END;
WRITELN;
WRITELN('STUDENTI S OTLICEN USPEH:');
BR:=0;
FOR I:= 1 TO N DO
 WITH ST[I] DO
 IF USPEH >= 5.50 THEN
         BEGIN
         BR:=BR+1;
         WRITELN(IME);
         END;
WRITELN('BROJ STUDENTI S OTLICEN USPEH:', BR);
END.
```

Въпроси

1. Какво е масив и какви са неговите характеристики?
2. Как се декларира масив?
3. Как се записва елемент на масив?
4. Какви са видовете масиви?
5. Дайте пример за работа с едномерен масив.
6. Дайте пример за работа с двумерен масив.
7. Какво е запис?
8. Какви са действията за работа с цял запис?
9. Какви са действията за работа с отделни полета от запис?