

§ 63. Табличні величини

Вивчивши цей параграф, ми:

познайомимося з табличними величинами;

дізнаємося, як описуються лінійні таблиці в алгоритмах і програмах;

з'ясуємо, як організується опрацювання лінійних табличних величин;

одержимо уявлення про багатовимірні таблиці.

====63.1. Таблиця та її елементи=====

У багатьох задачах виникає необхідність обробляти за певними правилами сукупність однотипних величин. Для того щоб можна було складати компактні алгоритми розв'язання таких задач й обробляти величини одну за одною в циклі, їх об'єднують у *масив*, або *таблицю*.

Таблиця — це поійменована сукупність однотипних величин із спільною властивістю, які розрізняються між собою своїм положенням у таблиці. Величини, які входять до таблиці, називаються її **елементами**.

Так, у таблицю можна об'єднати щоденні дані про температуру води на узбережжі Криму за літній сезон, результати спортивних змагань, оцінки учня з певного предмета за навчальний рік тощо.

Таблиці розрізняються за складністю їх організації. Найпростішими є таблиці, де елементи послідовно перенумеровані й упорядковані за їх номерами. Такі таблиці називають **лінійними**, або **одновимірними**. Номер величини в таблиці називають **індексом**. Кожний елемент таблиці має ім'я, спільне для всіх елементів таблиці, — ім'я таблиці і свій унікальний індекс. Індекс записують у квадратних дужках після імені таблиці. Елементи таблиці, на відміну від простих величин, називають *величинами з індексом*.

Наприклад, для опрацювання результатів змагань зі стрибків у висоту, де брали участь 100 спортсменів, утворимо таблицю, яка містить усі 100 результатів, показаних спортсменами. Ці результати упорядковані таким чином: спочатку результат, показаний першим учасником, потім другим і т. д. Надамо таблиці ім'я *rez*. Тоді *rez[1]* — це результат першого учасника, *rez[37]* — результат 37-го учасника, *rez[100]-rez[1]* — різниця між результатами останнього й першого учасників і т. ін.

Далі ми розглянемо, як в алгоритмах і програмах здійснюється опис і опрацювання таблиць. Як приклад таблиці будемо використовувати таблицю результатів спортивних змагань зі стрибків у висоту.

====63.2. Опис лінійних таблиць=====

Для того щоб мати можливість працювати з таблицею, її потрібно описати так само, як це здійснюється для простих величин. Опис таблиці містить вказівку про ім'я таблиці, а також тип і кількість величин, об'єднаних у таблицю. Цей опис дозволяє комп'ютеру визначити, який обсяг пам'яті потрібен для зберігання таблиці, виділити його й пов'язати з іменем таблиці.

Для опису таблиць у НАМ використовується службове слово таб. **Опис таблиці** здійснюється у такий спосіб:

```
тип_елементів таб ім'я_таблиці [найменший_індекс:найбільший_індекс]
```

Для нашого випадку опис таблиці має вигляд:

```
ціл таб rez [1:100]
```

оскільки результати стрибків у висоту подаються цілим числом сантиметрів.

У мові Паскаль таблиці звичайно називають масивами. Оголошення масиву здійснюється з використанням службових слів *array* (від англ. *array* — маса, множина) і *of* (із) у такий спосіб:

```
ім'я_масиву: array [ найменший_індекс..найбільший_індекс ] of тип_елементів;
```

Для нашого випадку опис таблиці мовою Паскаль має вигляд:

```
rez : array[1 .. 100] of integer;
```

За таким описом у пам'яті комп'ютера буде виділено 200 байт для розміщення таблиці, по 2 байти на кожний елемент.

Діапазон значень індексу можна задавати як за допомогою цілочисельних констант, так і за допомогою виразів, які дають значення цілого типу. Наприклад, наша таблиця може бути оголошеною як

```
rez : array[1 .. n] of integer;
```

але значення n у такому разі вже має бути визначеним раніше в розділі констант:

```
const n=100;
```

Такий спосіб визначення розміру таблиці є зручним, тому що можна складати текст програми для таблиці з n елементів, а для «налаштування» програми на конкретну таблицю змінювати тільки одне значення константи.

====63.3. Опрацювання лінійних таблиць=====

Опрацювання таблиць в алгоритмах і програмах звичайно описується за допомогою циклів із лічильником. У циклі здійснюється введення й виведення елементів таблиці, у циклі вони аналізуються або набувають нових значень. При складанні таких циклів дотримуються певних правил. У тілі циклу розміщують послідовність команд або операторів для опрацювання елемента таблиці, індекс якого задається змінною (наприклад, i), а в заголовку циклу вказують перелік значень індексу. Якщо опрацюванню підлягають усі без винятку елементи таблиці, то до цього переліку входить весь діапазон значень індексу — від найменшого до найбільшого значення (наприклад, для i від 1 до n). Отже, змінна, яка позначає індекс, виступає в ролі лічильника циклу. У нашому прикладі це змінна i .

Схему опрацювання всіх елементів таблиці $A[1 .. n]$ можна подати так:

Алгоритм у НАМ	Програма мовою Паскаль
<u>для</u> i <u>від</u> 1 <u>до</u> n	for $i:=1$ to n do
<u>пс</u>	begin
<команди опрацювання елемента $a[i]$ >	<оператори опрацювання елемента $a[i]$ >
<u>кс</u>	end;

Користуючись цією схемою, запишемо алгоритм підрахування середнього результату, показаного спортсменами на змаганні. Для обчислення середнього результату (sr) потрібно знайти суму всіх результатів (s) і поділити її на кількість учасників змагань (n).

Для того щоб уникнути введення з клавіатури елементів таблиці (результатів, показаних спортсменами), надамо їм випадкових значень із діапазону [170; 199].

Алгоритм у НАМ	Програма мовою Паскаль
<u>алг</u> середнє (<u>ціл</u> n ; <u>ціл</u> <u>таб</u> <u>рез</u> [1: n]; <u>дійс</u> sr) <u>арг</u> n , <u>рез</u> <u>рез</u> sr	program average; uses Crt; const $n=100$; var i , s : integer ; sr : real ;

	<code>rez: array [1 .. n] of integer;</code>
<u>нач</u> <u>ціл</u> <i>i</i> , <i>s</i>	begin
	for <i>i</i> :=1 to <i>n</i> do
	<code>rez[<i>i</i>] := 170+random(30);</code>
<code><i>s</i> := 0</code>	<code><i>s</i> := 0;</code>
<u>для</u> <i>i</i> <u>від</u> 1 <u>до</u> <i>n</i>	for <i>i</i> :=1 to <i>n</i> do
<code><i>s</i> := <i>s</i>+rez[<i>i</i>]</code>	<code><i>s</i> := <i>s</i>+rez[<i>i</i>];</code>
<code><i>sr</i> := <i>s</i>/<i>n</i></code>	<code><i>sr</i> := <i>s</i>/<i>n</i>;</code>
	writeln (<i>sr</i> :5:1);
	readln
<u>кін</u>	end.

====63.4. Знаходження найменшого елемента таблиці=====

У процесі розв'язання багатьох задач опрацювання таблиць виникає потреба знайти найменший або найбільший елемент серед усіх елементів таблиці. Розглянемо, як здійснюється пошук найменшого елемента.

Надамо таблиці з *n* елементів ім'я *a*, позначимо шукане значення її найменшого елемента через *min*.

Для визначення найменшого елемента таблиці застосовується такий спосіб:

- надаємо змінній *min* значення першого елемента таблиці;
- порівнюємо *min* з другим елементом таблиці. Якщо він менший за *min*, то надаємо *min* значення другого елемента. Тепер змінна *min* містить значення меншого з двох елементів таблиці;
- порівнюємо *min* з третім елементом таблиці. Якщо він менший за *min*, то надаємо *min* значення третього елемента. Тепер змінна *min* містить значення найменшого з трьох елементів таблиці;
- просуваємося у такий спосіб до кінця таблиці і врешті-решт отримуємо значення *min* — найменшого з усіх елементів таблиці.

Описані дії можна подати у вигляді циклу:

Подання у НАМ	Подання мовою Паскаль
<code><i>min</i> := <i>a</i>[1]</code>	<code><i>min</i> := <i>a</i>[1];</code>
<u>для</u> <i>i</i> <u>від</u> 2 <u>до</u> <i>n</i>	for <i>i</i> :=2 to <i>n</i> do
<u>якщо</u> <code><i>a</i>[<i>i</i>]<<i>min</i></code>	if <code><i>a</i>[<i>i</i>]<<i>min</i></code>
<u>то</u> <code><i>min</i> := <i>a</i>[<i>i</i>]</code>	then <code><i>min</i> := <i>a</i>[<i>i</i>];</code>

Якщо в наведеному фрагменті знак порівняння «<» замінити на «>», то буде знайдено не


```
writeln('Переможець змагань - ', pp);
readln;
end.
```

====63.6. Поняття про багатовимірні таблиці=====

Лінійні таблиці є найпростішою формою організації даних. Pozнайомимося з тим, як утворюються *багатовимірні таблиці*, де кожний елемент визначається не одним, а декількома індексами.

Нехай таблиця *w* містить оцінки певного учня (наприклад, Петрова) за навчальний рік з 10 предметів: української мови, математики, інформатики тощо. Тоді *w[1]* — оцінка Петрова з української мови, *w[2]* — з математики тощо. Таблиця *w* описується як *w*:

```
array[1 .. 10] of integer.
```

Аналогічні таблиці для всіх 25 учнів класу (Альоxін, Барахтян, ..., Петров, ..., Яшин) зведемо в одну таблицю *ww*. У першому її рядку розмістимо оцінки Альоxіна, у другому — Барахтяна і т. д. Таблиця *ww* є двовимірною. Вона складається з 25 рядків, по 10 елементів у кожному, і описується як *ww*: **array[1..25,1..10] of integer**. Кожен з 250 елементів таблиці визначається двома індексами. Наприклад, для того щоб дізнатися про оцінку Петрова з математики, потрібно вказати номер, за яким йде Петров у списку учнів (наприклад, 17), і номер, за яким йде математика у списку предметів (номер 2). Отже, оцінка Петрова з математики дістається як *ww[17, 2]*.

Можна уявити собі й тривимірну таблицю — об'єднання оцінок учнів паралельних класів — і т. д.

ВИСНОВКИ

Для опрацювання однотипних величин за певним правилом їх об'єднують у таблиці. Окремі величини, які входять у таблицю, називають її елементами. Кожний елемент має своє місце в таблиці. Таблиці надається ім'я. Це ім'я є спільним іменем усіх елементів таблиці. Для вказівки на певний елемент таблиці потрібно зазначити ім'я таблиці й місце елемента в таблиці. Місце елемента таблиці задається індексами, які розміщуються у квадратних дужках після імені таблиці. Таблиці найпростішого виду — лінійні. Вони містять перенумеровану сукупність елементів. Щоб задати певний елемент таблиці, вказують його номер. Використання таблиць дозволяє складати компактні програми опрацювання великих масивів даних.

Контрольні питання та вправи

1. Лінійна таблиця об'єднує ... елементи. Кожний елемент має своє ... в таблиці. Усі елементи мають ... імена. Для розрізнення елементів до імені додається

- а) лінійні;
- б) однакові;
- в) однотипні;
- г) різні за типом;
- д) індекс;
- е) місце;
- є) значення;
- ж) масив.

2. Виберіть оголошення масиву, які не містять помилок.

- а) `army: array [1 : 10] of integer;`
- б) `z: array of [0 .. 41] integer;`
- в) `s: integer of array [1 .. 8];`

г) class11: array [1 .. 25] of string;
 д) price: array [1 .. 1000] of real;
 е) w: array of integer [1 .. 10].

3. Укажіть, які значення будуть надані елементам масиву в результаті виконання оператора:

1) for k:= 1 to 5 do
 if k>3 then a[k]:=k else a[k]:=7;
 2) for k:=1 to 5 do
 if k<>3 then a[k]:=k else a[k]:=7;
 3) for k:=1 to 5 do
 if k mod 3 = 2 then a[k]:=k else a[k]:=7;
 4) for k:=1 to 5 do
 if k mod 3 < k then a[k]:=k else a[k]:=7.

а) 7, 7, 7, 4, 5;
 б) 1, 2, 3, 4, 5;
 в) 1, 2, 3, 7, 7;
 г) 7, 7, 3, 4, 5;
 д) 7, 2, 7, 7, 5;
 е) 7, 2, 7, 4, 7;
 є) 1, 2, 7, 4, 5;
 ж) 1, 2, 7, 7, 7.

4. Лінійна таблиця $a[1:n]$, де $n=7$, містить елементи:

18, 24, 6, 48, 0, 12, 6.

Знайдіть значення елементів таблиці після виконання фрагмента алгоритму:

1) для k від 1 до n
 a[k] := a[n-k+1]
 2) для k від 1 до n
 a[k] := a[k] / a[n-4]
 3) t := a[1]
 для k від 1 до n-1
 a[k+1] := a[k]
 a[1] := t
 4) t := a[n]
 для k від n до 2 крок -1
 a[k] := a[k-1]
 a[1] := t

5. Дана таблиця, яка містить посадові оклади n працівників фірми. Складіть програму:

1) визначення середнього посадового окладу працівника фірми;
 2) підвищення окладів усіх працівників фірми на z грн.

6. Дана таблиця, яка містить щоденні показники про середню температуру повітря впродовж березня. Складіть програму підрахунку, яких днів у березні було більше — з температурою вище чи нижче нуля.

7. Утворіть дві таблиці: таблицю t , яка містить список товарів у магазині, і таблицю s , яка містить їх ціни. Складіть програму, яка виводить на екран список товарів, що коштують дешевше заданого z .

8. Складіть програму, яка моделює телевізійне шоу «Щасливий випадок». Перед гравцем клітинне поле, яке містить 5×5 клітинок. Гравець робить «постріл» (уводить номери клітинки по горизонталі і по вертикалі). За певними клітинками сховані подарунки: запрошення у романтичну подорож, вечеря в ресторані «Престиж», комплект парфумів тощо.

9. Складіть програму-тренажер, яка пропонує користувачеві перевірити його знання столиць відомих країн. Програма одну за одною виводить на екран назви країн, а користувач уводить із клавіатури назви їх столиць. Програма перевіряє правильність кожної відповіді й на завершення виводить повідомлення про результати перевірки.

10. Складіть програму «Іподром». Програма виводить на екран список учасників забігу й пропонує користувачеві зробити ставку на певного учасника. Результати забігу визначаються за допомогою датчика випадкових чисел. Програма виводить їх на екран, визначає переможця і повідомляє користувача, програла чи виграла його ставка.

11. Складіть програму «Касир». Програма за даними про вартість покупки й надану покупцем суму грошей визначає, скільки та яких купюр потрібно видати йому на здачу. Здача дається найменшою кількістю купюр.

12. Складіть програму-тренажер, яка пропонує користувачеві перевірити його знання з англійської мови. Програма виводить на екран слово української мови, а користувач із клавіатури вводить переклад цього слова на англійську. Програма перевіряє правильність кожної відповіді й на завершення виводить повідомлення про результати перевірки. Удоскональте програму так, щоб вона за вибором користувача перевіряла переклад з англійської на українську, або з української на англійську, або довільно вибирала слова то однієї, то іншої мови.

багатовимірна таблиця, елемент таблиці, індекс, лінійна таблиця, масив, оголошення масиву, одновимірна таблиця, опис таблиці, таблиця