**Тема. Масиви. Одновимірні масиви**

*Масив* - структурований тип даних, що складається з фіксованого числа елементів одного типу.

Масив, що представляє собою просто список даних одного і того ж типу, називають **простим, або одновимірним масивом**.

**Для доступу до даних**, що зберігаються в певному елементі масиву, необхідно вказати ім'я масиву і порядковий номер цього елемента, званий індексом.

Кожен масив має ім'я і розмірність:

• ім'я масиву - це будь-яка буква латинського алфавіту.  
• розмірність масиву - кількість елементів цього масиву.

У загальному вигляді інструкція оголошення масиву виглядає таким чином:

**Ім'я: array [нижній індекс. .** **верхній індекс] of тип**

де: - ім'я - ім'я масиву;

- array - зарезервоване слово мови Delphi, що позначає, що оголошуване ім'я є ім'ям масиву;

- нижній індекс і верхній індекс - цілі константи, що визначають діапазон зміни індексу елементів масиву і, неявно, кількість елементів (розмір) масиву;

- тип - тип елементів масиву.

**var A : array[ 1 .. 5 ] of integer** ;

Описати масив можна наступними способами.

1. **type**  
   Ім’я\_Типу = **array** [тип\_індекса] **of** тип\_компонентів;

наприклад:

**type**  
massiv = array [0 .. 12] **of** real;  
// *Тип даних massiv з 13 елементів, які нумеруються від 0 до 12.*dabc = array [- 3 .. 6] of integer;  
// *Тип даних dabc з 10 елементів, які нумеруються від* - *3 до 6.*  
**var**  
x, y: massiv;  
z: dabc;

1. **var** змінна: **array** [тіп\_індекса] **of** тип\_змінних;

Наприклад:  
**var**  
z, x: **array** [1 .. 25] **of** word; // *Масиви z і x з 25 значень типу word, які нумеруються від 1 до 25*.  
g: **array** [- 3 .. 7] **of** real; // *Масив g з 11 значень типу real, які нумеруються від -3 до 7.*

1. **const**  
   n = 10;  
   m = 12;  
   **var**  
   a: **array** [1 .. n] **of** real;  
   b: **array** [0..m] **of** byte;

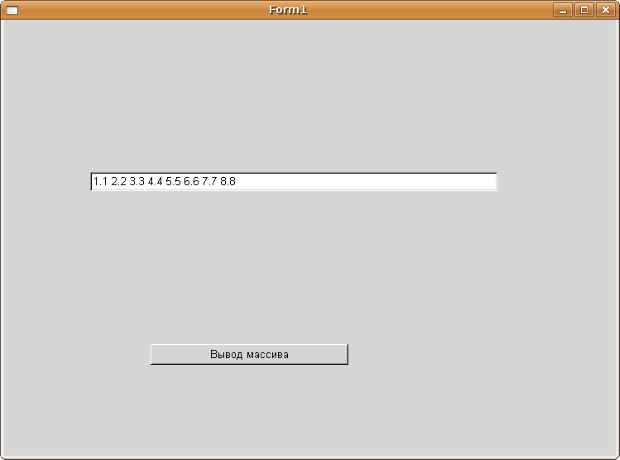
Константи повинні бути визначені до використання, так як масив не може бути змінної довжини!

Завдання з програмування з використанням масивів вирішуються, як правило, за наступним алгоритмом: оголошення масиву, генерація або ініціалізація, обробка значень елементів, виведення.

Виведення даних в консольному режимі

// *Введення елементів масиву X за допомогою циклу* **while***.*  
**var** x: **array** [1 .. 100] **of** real;  
i, n: integer;  
**begin**  
**writeln** ('введіть розмір масиву');  
**readln** (N);  
i: = 1;  
**while** (i <= N) do  
**begin**  
**write** ('x (', i, ') ='); **readln** (x [i]);  
i: = i +1  
**end; end;**  
// *Введення елементів масиву X за допомогою циклу* **for**.  
**var**  
x: **array** [1 .. 1 0 0] **of** real;  
i, n: integer;  
**begin**  
**readln** (N);  
**for** i: = 1 **to** N **do**  
**begin**  
**write** ('x (', i, ') ='); **readln** (x [i])  
**end**;  
**end**.

Виведення даних в візуальних додатках

**procedure TForm1. Button1Click (Sender: TObject);  
var**  
a: **array** [1 .. 8] **of** real = (1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8); // *Вихідний масив a.*  
i: integer;  
n: integer = 8; // *У змінній n зберігається кількість елементів у масиві* *дійсних чисел.*   
S: string = ' ';  
**begin**  
Edit1. Text: = ' ';  
**for** i: = 1 **to** n **do**// *Цикл for для послідовної обробки всіх елементів* *масиву a.*  
// *Додавання в полі введення Edit1 рядки, які отримані з елементу масиву a [i].*Edit1. Text: = Edit1. Text + FloatToStr (a [i]) + ' ';  
**end**;

Введення розміру масиву та елементів масиву в поле InputBox

**procedure TForm1. Button1Click (Sender: TObject);  
var** i, n: byte;

X: **array** [1 .. 20] **of** real;  
**begin**  
// *Кількість елементів масиву*.  
n: = StrToInt (InputBox ("Введення елементів масиву ',' n = ','7'));  
**for** i: = 1 **to** n **do** // *Поелементне введення.*  
// *Введення чергового елементу масиву*.  
X [i]: = StrToFloat (InputBox ("Введення елементів масиву ',  
'Введіть' + IntToStr (i) + 'елемент', '0,00'));  
**end**;

Приклад програми. Дано масив X, що складається з n елементів. Знайти суму елементів цього масиву.  **const n=3;**

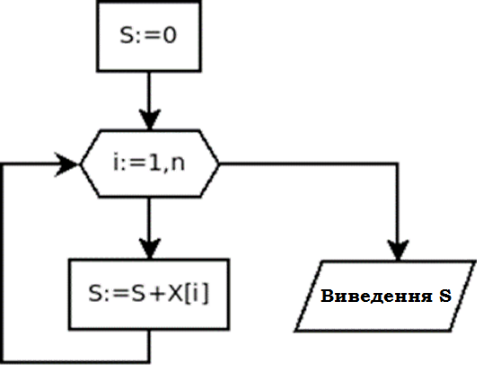
x: **array** [1 .. 100] **of** real; //*опис одновимірного масиву, де real- дійсний тип*

s: **real**; //*змінна дійсного типу*

i: **integer**; //*змінна цілого типу*

**begin**

**for** i:= 1 **to** n **do //***введення даних у масив*

 **begin**

write ('x (', i, ') =');

readln (x[i]); //*вводяться елементи масиву*

**end**;

s:= 0; //змінній s *привласнюємо початкове значення*

**for** i:= 1 **to** n **do**

s:= s + x [i]; //*підраховуємо суму елементів масиву*

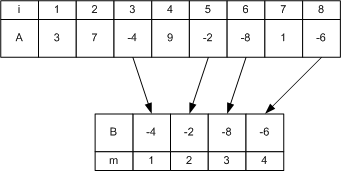
writeln ('s =', s: 7: 3); //*виведення суми елементів масиву*

readln();

**end**.

**Домашнє завдання**

**СВ:** Розглянути компонент String Grid. ДЗ виконати за допомогою даного компонента

**Високий рівень** ЗАВДАННЯ. Знайти суму додатніх елементів послідовності d1, d2, ..., d40, розташованих до першого нульового елемента, замінити цією сумою мінімальний елемент масиву.

**Середній рівень** ЗАВДАННЯ. Дан масив А, що складається з k цілих чисел. Записати всі негативні елементи масиву A в масив B.

**Достатній рівень** ЗАВДАННЯ. В масиві а1, а2, ..., а50 визначити кількість нулів.