

Построение структур данных



Простейшие данные

Структурированные данные

1. **Простейшие данные** – элементы данных, являющиеся неделимыми (числа, строки, знаки). Для простейших данных существуют стандартные типы (Integer, Real, Char, String, Boolean).
2. **Структурированные данные** – это структуры, состоящие из нескольких простейших данных. Определяются пользователем в программе при помощи двух конструкций:

Массивы, записи

- Массив – структура однотипных данных с индексированным доступом.
- Запись – структура данных с доступом по идентификатору.

Определение

- Массив – структура однотипных данных с индексированным доступом. Каждый элемент массива получает один или несколько номеров, называемых индексами. Индексы записываются в квадратных скобках через запятую.

Виды массивов

- **Одномерные** – каждый элемент массива получает два индекса (пр. [2,3]).
- **Многомерные** – каждый элемент получает более 2-х индексов (пр. [1,1,k]).

Описание массивов

- Массив в языке Паскаль это сложный тип данных, поэтому чаще всего его описывают в разделе переменных.
- `Var <переем. массив>: array[<диапазон 1>..<диапазон N>]`
- `Of <тип переменной>;`

Получение элементов

массива

- Каждый элемент массива — это переменная, которой можно присваивать значения в операторах и функциях. Для того, чтобы указать элемент массива, необходимо записать все его индексы.
- `<перем. массив> [<индекс>, ..., <индекс N>]`
- Пример:
- `Spisok[1] := 'Иванов';`

Ввод массива с клавиатуры

- Массив нельзя ввести с клавиатуры одной командой, для этого организуется цикл с параметром. Для того чтобы обеспечить правильный ввод массива, необходимо соблюдать следующий алгоритм:
- <приглашение к вводу массива>;
- <ввод количества элементов массива>;
- For i:=1 to <кол-во элементов> do
- Begin
- <приглашение к вводу i-го элемента>;
- <ввод i-го элемента>;
- End;

Пример:

- WriteLn (^ввод списка студентов');
- WriteLn (^введите количество студентов');
- ReadLn (kolvo);
- For i:=1 to kolvo do
- Begin
- WriteLn (i, ' ');
- ReadLn (spisok[i]);
- End;

Вывод массива на экран

Вывод в строку

1. При выводе массива в строку нужно использовать `Write`, которая будет находиться в цикле с параметром, а после цикла нужно поставить `WriteLn`.

Пример:

- For i:=1 to kolvo do
- Write(spisok[i], ' ');
- WriteLn;

Вывод массива в столбец.

- При выводе в столбец в цикле указывается `WriteLn`, причем для массивов, содержащих числа, следует также указывать количество знаков при выводе на экран.

Пример:

- For i:=1 to kolvo do
- Write (i:2, '. ',spisok[i]);

Действия с массивами

- Над массивами нельзя выполнять арифметические действия (вычитать, складывать и др.). Все действия выполняются поэлементно.

Пример:

- Написать программу, которая выполняет поэлементное суммирование массивов А и В.

- Program Summ;
- Var A,B Array [1..10] of Integer;
- Kolvo:Integer;
- C: Array [1..10] of Integer;
- i: Integer;
- Begin
- WriteLn (^введите количество элементов массивов^);
- ReadLn (Kolvo);
- WriteLn (^введите элементы массива A^);
- For i:=1 to Kolvo do

- begin
- Write(i, ` `);
- ReadLn (A[i]);
- end;
- WriteLn (^введите элементы массива B^);
- For i:=1 to Kolvo do
- begin
- Write(i, ` `);
- ReadLn (B[i]);
- end;
- For i:=1 to Kolvo do
- C[i]:= A[i]+B[i];
- For i:=1 to Kolvo do
- Writeln (C[i]:5);
- End.

Двухмерные массивы

Ввод построчно

- <приглашение ввода массива>;
- <цикл по строкам>;
- Begin
- <приглашение ввода строки>;
- <цикл по столбцам>;
- Begin
- <приглашение ввода элемента>;
- <ввод элемента массива>;
- End;
- End.

Ввод по столбцам

- <приглашение ввода массива>;
- <цикл по столбцам>;
- Begin
- <приглашение ввода столбца>;
- <цикл по строкам>;
- Begin
- <приглашение ввода элемента>;
- <ввод элемента массива>;
- End;
- End

Пример

- **Задача:**
- В одномерном массиве найти максимальный из отрицательных элементов и поменять его местами с последним элементом массива.

Описание переменных:

- N - количество элементов в массиве;
- a - массив;
- i - счетчик элементов;
- j - индекс максимального отрицательного элемента;
- b - "буферная" переменная для обмена.

Алгоритм решения задачи:

- Присвоить j ноль. Это будет сигналом о том, что отрицательных элементов вообще нет. Если очередной элемент массива меньше нуля и $j = 0$, то присвоить его индекс переменной j . Если он отрицательный, а j не равно 0, то если он больше, чем значение под j , переписать j на текущий индекс. Если значение j не равно нулю, произвести обмен и вывести массив.

Программа на языке Паскаль:

- **const** N = 15; **var** a:
array[1..N] **of** **integer**;
i, j: **byte**; b: **integer**;
begin randomize; **for**
i:=1 **to** N **do** **begin** a[i]
:= random(100)-50;
write(a[i], ' '); **end**;
writeln; j := 0; **for** i:=1
to N **do** **if** a[i] < 0 **then**
if j = 0 **then** j := i **else**
if a[i] > a[j] **then** j := i;
- **if** j <> 0 **then** **begin** b
:= a[j]; a[j] := a[N]; a[N]
:= b; **for** i:=1 **to** N **do**
write(a[i], ' '); writeln;
end **else**
writeln('Отрицательных
элементов нет'); **end**.

Пояснения к программе

- Процедура **Randomize** используется вместе с функцией **Random**. Она устанавливает генератор случайных чисел в его последовательность из 232 псевдослучайных чисел

Randomize

- **Randomize** для этого повторного установления использует время дня как начальное значение, обеспечивая, таким образом, надежный метод создания непредсказуемой последовательности чисел, даже если они часть predetermined последовательности.

Похожие команды

- **Random** Генерирует случайное целое число или число с плавающей запятой

RandomRange Генерирует произвольное число в пределах введённого диапазона

RandSeed Устанавливает генератор случайного числа на следующее значение

Сортировка через два массива

- **Задача:**
- Даны два массива с различным количеством элементов.
Перераспределить их элементы так, чтобы в первом массиве были наименьшие из двух массивов, а во втором - наибольшие.

Алгоритм решения задачи:

- Задача решается сортировкой через два массива. В коде ниже выбран метод пузырька.
- В результате в одном массиве оказываются наименьшие элементы, а во втором - наибольшие. Причем в каждом массиве элементы отсортированы по возрастанию.

Программа на языке Паскаль:

- **const** M = 20; N = 15; **var** a1:
array[1..M] **of** **word**; a2: **array**[1..N] **of**
word; i, j, k: **byte**;
- **begin** randomize; write ('Исходный
массив №1: '); **for** i := 1 **to** M **do** **begin**
a1[i] := random(256); write (a1[i], ' ');
end;

Продолжение программы1

- writeln; write ('Исходный массив №2: ');
for i := 1 **to** N **do begin** a2[i] :=
random(256); write (a2[i], ' '); **end**;
writeln; **for** i := 1 **to** M+N-1 **do begin**
for j := 1 **to** M+N-i **do begin**
■ **if** j < M **then begin** **if** a1[j] > a1[j+1]
then begin k := a1[j]; a1[j] := a1[j+1];
a1[j+1] := k **end**; **end**

Продолжение программы2

- **else begin if j = M then begin if a1[j] > a2[1] then begin k := a1[j]; a1[j] := a2[1]; a2[1] := k end; end else begin k := a2[j-M]; a2[j-M] := a2[j-M+1]; a2[j-M+1] := k end; end; end; end;**
- **write ('Отсортированный №1: '); for i := 1 to M do write (a1[i], ' '); writeln; write ('Отсортированный №2: '); for i := 1 to N do write (a2[i], ' '); writeln; end.**