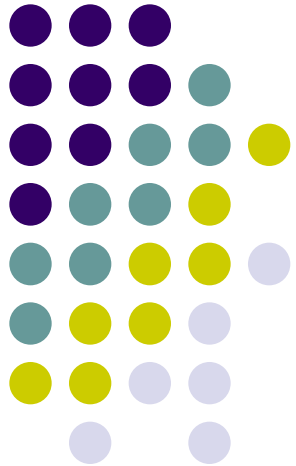
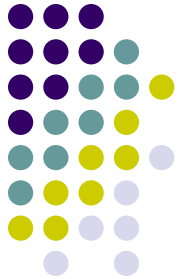
Програмування в Delphi: масиви

1 Одновимірні масиви
1.1 Оголошення масиву
1.2 Виведення масиву
2 Багатовимірні масиви
3 Помилки при використанні масивів



Масив – це структура даних, доступ до елементів якої здійснюється за номером індексу. Всі елементи масиву мають один тип.

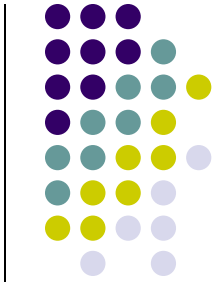


- **1 Одновимірні масиви**
- **1.1 Оголошення масиву**

Масив, як і будь-яка змінна програми, перед використанням повинен бути оголошений в розділі змінних.

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

Приклади оголошення масивів:



- *temper:array[1..31] of real;*
- *koef:array[0. .2] of integer;*
- *name:array[1..30] of string[25];*

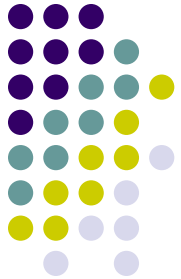
При оголошенні масиву зручно використовувати іменовані константи перед розділом оголошення змінних.

- *const*
- *NT = 18; // число груп*
- *SN = 25; // гранична довжина назви групи*

var

team: array[1..NT] of string[SN];

Для того, щоб в програмі використовувати елемент масиву, треба вказати ім'я масиву і номер елементу (індекс). Як індекс можна використовувати константу або вираз цілого типу, наприклад:



- team [1] := ‘прикладна';
- d := koef[l]*koef[l]-4*koef[2]*koef[1];
- ShowMessage(name[n+1]);
- temper[i]:= StrToFloat(Edit1.text);

Оголошення масиву з одночасною його ініціалізацією в загальному вигляді :

a: array[10] of integer = (0,0,0,0,0,0,0,0,0,0);

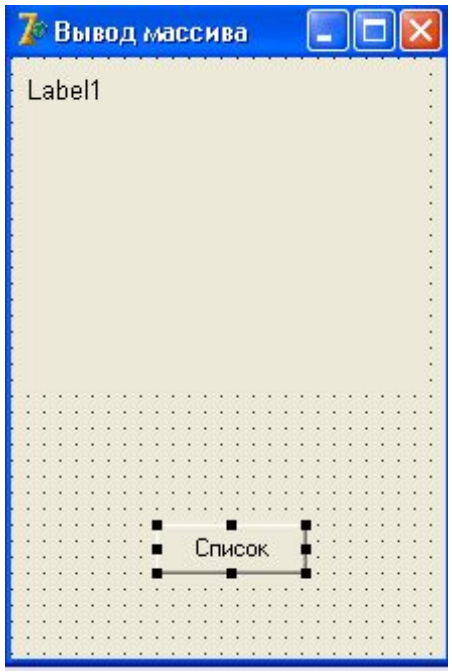
Team: array[1..5] of String[10]=

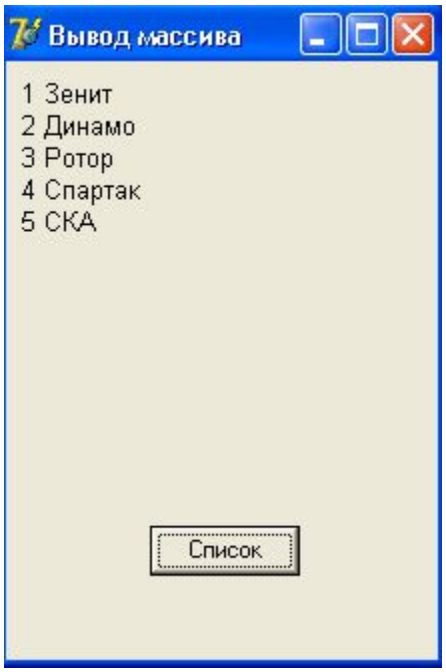
('Зенит','Динамо','Спартак','Ротор','СКА');

Кількість елементів списку ініціалізації повинна відповідати розмірності масиву. Інакше компілятор виведе повідомлення про помилку: **Number of elements differs from declaration** (кількість елементів не відповідає вказаному в оголошенні).

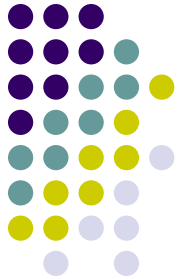
1.2 Виведення масиву

- Label1.AutoSize := False;



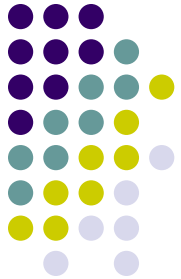


Лістинг 1. Ініціалізація і виведення масиву



```
const
NT = 5;
var
team: array[1..NT] of string[10]=('Зенит','Динамо','Ротор','Спартак','СКА');
procedure TForml.ButtonlClick(Sender: TObject);
var
st:string; // список команд
i:integer; // індекс, номер елементу масиву
begin
// формування списку для відображення у формі
for i:=l to NT do
 st := st + IntToStr(i)+ ' ' + team[i]; // виведення списку
 Label1.Caption := st;
end;
end.
```

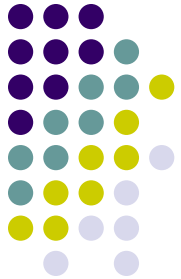
2 Багатовимірні масиви



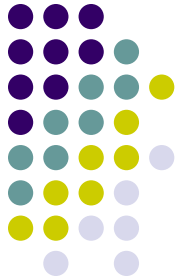
Наприклад, результат діяльності деякої фірми, торгуючої автомобілями, може бути представлений у вигляді таблиці

	січень	лютий	…	грудень
марка1				
марка2				
марка3				
марка4				

приведена таблиця може бути представлена як сукупність одновимірних масивів



- marka1: array [1..12] of integer;
- marka2 : array [1..12] of integer;
- marka3 : array [1..12] of integer;
- marka4 : array [1..12] of integer;

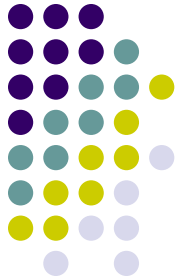


В цьому випадку кожен масив призначений для зберігання інформації про кількість проданих за місяць автомобілів, причому значення елементу масиву відображає продану кількість автомобілів однієї марки.

- jan: array [1..4] of integer;
- feb: array [1..4] of integer;
- mar: array [1..4] of integer;
- dec: array [1..4] of integer;

Якщо вся таблиця містить однорідну інформацію, наприклад, тільки цілі числа, то така таблиця може бути представлена як двовимірний масив.

оголошення двовимірного масиву



- Ім'я: array[Нижняя Межа1..Верхняя Межа 1; Нижняя Межа 2.. Верхняя Межа 2] of Тип
- rezult: array [1..12, 1..4] of integer

Наступний фрагмент програми обчислює суму елементів масиву (загальна кількість автомобілів, проданих за рік).

```
s:=0;
for i := 1 to 4 do // 4 моделі автомобілів
for j := 1 to 12 do //12 місяців
s := s + rezult[i,j];
```

3 Помилки при використанні масивів

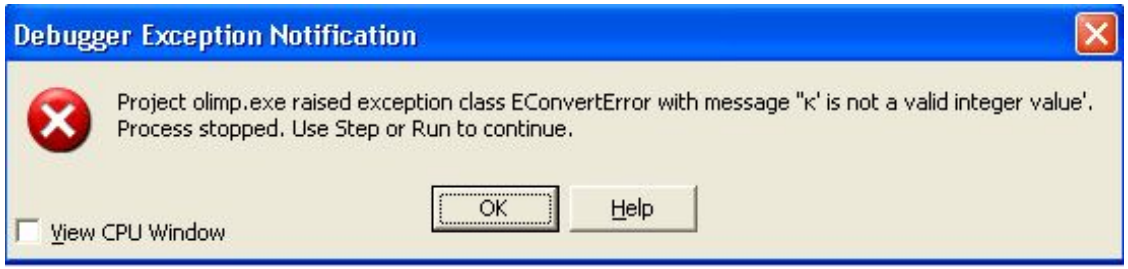
Якщо **tab1: array [1..N] of integer;**

то інструкція

for i:=0 to N do tab1[i] := 5;

формально є вірною, і її компіляція буде успішно виконана.

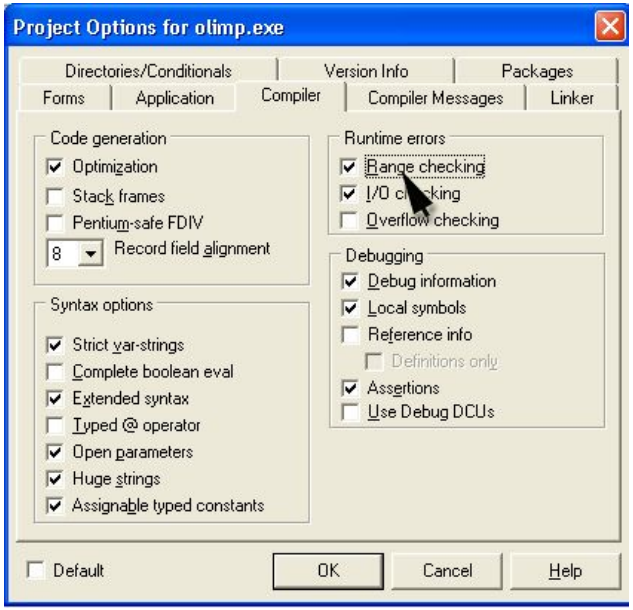
- Проте під час виконання програми, при спробі привласнити значення неіснуючому нульовому елементу масиву tab, на екран буде виведене повідомлення про помилку при зверненні до неіснуючого елементу масиву



Повідомлення про помилку при зверненні до неіснуючого елементу масиву

- При спробі привласнити значення неіснуючому елементу масиву на екран буде виведене повідомлення Range check error (помилка контролю діапазону). У заголовку вікна буде вказане ім'я застосування, в процесі виконання якого відбулася помилка.

Поведінка програми при виході індексного виразу за межі діапазону допустимих значень визначається налаштуванням компілятора

- Для того, щоб програма контролювала значення індексних виразів (в цьому випадку Delphi додає у виконувану програму інструкції, що забезпечують цей контроль), необхідно на вкладці Compiler діалогового вікна Project Options, яке відкривається вибором з меню Project команди Options, встановити прапорець Range checking (Контроль діапазону), що знаходиться в групі

  Runtime errors (Помилки часу виконання)