

Лекція 5

Масиви



План на сьогодні

1

Що таке масив?

2

Оголошення масиву

3

Ініціалізація масиву

4

Оператор sizeof()

5

Багатовимірні масиви



FACULTY OF APPLIED
MATHEMATICS AND
INFORMATICS
LVIV UNIVERSITY

Що таке масив?

Що таке масив?

Елемент одного
типу

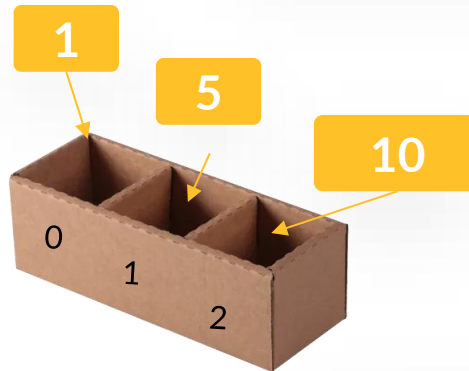
```
int a = 5;
```



Змінна
“a”

Масив

```
int b[3] = {1, 5, 10};
```



Змінна
“b”

Що таке масив?

Масив — впорядкований набір елементів **одного типу**, доступ до яких здійснюється за індексом — порядковим номером елемента в масиві

Визначення масиву

Для визначення масиву необхідно вказати:

1. Тип елементів масиву (**dataType**)
2. Назву масиву (**arrayName**)
3. Розмір масиву (кількість елементів) (**arraySize**)



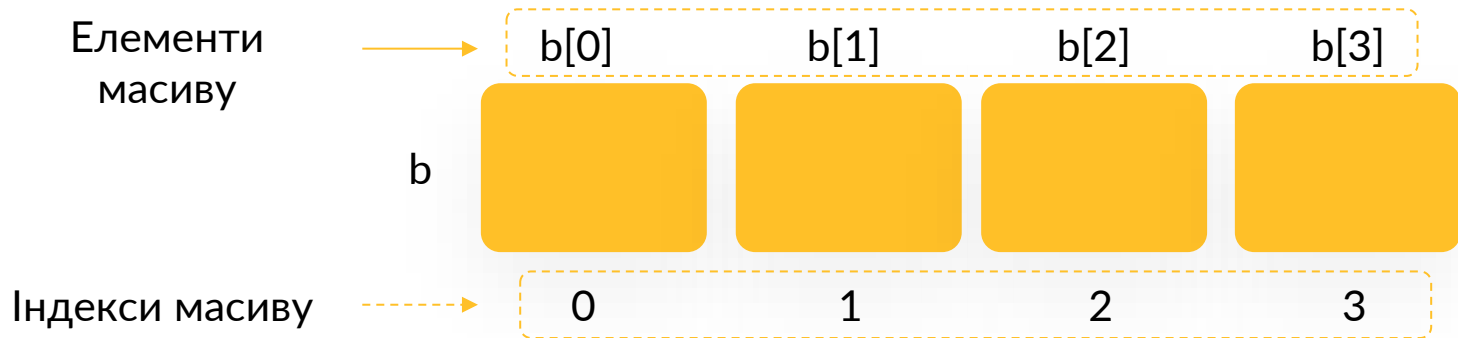
```
dataType arrayName[arraySize];
```

Розмір масиву незмінний протягом життя програми,
повинен бути константою

Визначення масиву

```
int b[4];
```

- int - тип елементів, які будуть зберігатися в масиві
- b - назва масиву
- 4 - розмір масиву



Приклади оголошення масивів

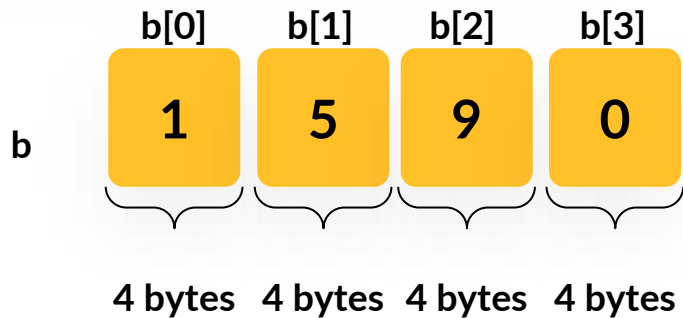
```
1. const int size = 512, M = 20;  
2. int N = 27;  
3. char arr[size];  
4. char fileT[M-3];  
5. double mas[13];  
6. double salaries[N]; // помилка N – не константа
```


Приклади оголошення з ініціалізацією

```
1. float array[] = { 0.3, 1.3, 2.8}; // масив розміру 3
2. const int M = 5;
3. int array2[M] = { 0, 1, 2}; // mas2: { 0, 1, 2, 0, 0 }
4. const char letters[] = {'C', '+', '+'}; // розмір 3
5. const char letters2[] = "C++"; // розмір 4, c-style string + '\0'
6. const char letters3[6] = "Daniel"; // помилка
```

Доступ до елементів через індекси

```
int b[4] = {1, 5, 9, 0};
```



До елементів масиву доступ здійснюється через оператор `[]` (індексування):

- `b[0]`; // перший елемент
- `b[3]`; // останній елемент
- `b[4]`; // помилка

C++ не забезпечує контроль індексів масиву – ні на етапі компіляції, ні виконання

Операції над елементами масиву

1. `int mark[5] = {19, 10, 8, 17, 9}`
2. `// змінити 4-ий елемент на число 7`
3. `mark[3] = 7;`
4. `// ввести число і зберегти в масиві, в 3-ій комірці`
5. `cin >> mark[2];`
6. `// ввести число і зберегти в масиві, в i-ій комірці`
7. `int i=0`
8. `cin >> mark[i-1];`
9. `// видрукувати перший елемент масиву`
10. `cout << mark[0];`
11. `// знайти суму першого і останнього елементів і видрукувати`
12. `cout >> mark[0]+mark[4];`

Приклади

Приклад 1

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string cars[5] = {"Volvo", "BMW", "Ford", "Mazda", "Tesla"};

    for (int i = 0; i < 5; i++) {
        cout << cars[i] << "\n";
    }
    return 0;
}
```

```
Volvo
BMW
Ford
Mazda
Tesla
```

Приклад 2. Range-based for loop (since C++11)

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     // Create an array of strings
7     string cars[5] = { "Volvo", "BMW", "Ford", "Mazda", "Tesla" };
8
9     // Loop through strings
10    for (const string& car : cars) {
11        cout << car << "\n";
12    }
13    return 0;
14 }
```

```
Volvo
BMW
Ford
Mazda
Tesla
```

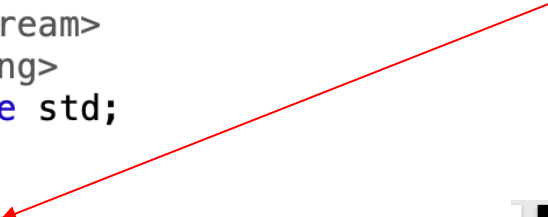
- **Зауваження:** чому `const string& car` замість `string car` оскільки:
- `string car` робить копіювання елементів масиву в змінну `car` під час кожної ітерації. Це не ефективно. `&car`, використовує адресу кожного елемента в пам'яті для доступу (без копіювання до нової змінної).
- Оскільки в циклі ми не змінюємо елементи масиву (тільки друкуємо), тому вказуємо `const`.

Приклад 3. Помилка

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    string cars[];
    cars[0] = "Volvo";
    cars[1] = "BMW";
    cars[2] = "Ford";
    cars[3] = "Mazda";
    cars[4] = "Tesla";
    for(int i = 0; i < 5; i++) {
        cout << cars[i] << "\n";
    }
    return 0;
}
```

Не вказаний розмір масиву



```
prog.cpp: In function 'int main()':
prog.cpp:6:10: error: array size missing in 'cars'
     6 |     string cars[];
       |           ^~~~
```

Приклад 4. Введення з клавіатури

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string names[3];

    cout << "Input names: " << endl;

    for(int i = 0; i < 3; i++) {
        cin >> names[i];
    }

    for (int i = 0; i < 3; i++) {
        cout << names[i] << "\n";
    }
    return 0;
}
```

Input names:

Оператор sizeof();

Приклад 4. Оператор sizeof()

```
#include <iostream>
using namespace std;

int main() {
    int myNumbers[5] = {10, 20, 30, 40, 50};
    cout << sizeof(myNumbers);
    return 0;
}
```

20

sizeof() - Повертає розмір в пам'яті цієї змінної. В даному випадку цілого масиву.

Приклад 5. Кількість елементів.

```
#include <iostream>
using namespace std;

int main() {
    int myNumbers[5] = {10, 20, 30, 40, 50};
    int getArrayLength = sizeof(myNumbers) / sizeof(myNumbers[0]);
    cout << getArrayLength;
    return 0;
}
```

Приклад 6. Середнє арифметичне.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      double numbers[] = { 7.19, 5.26, 6.33, 12.01, 35.15, 27.71 };
6      const size_t size = sizeof(numbers) / sizeof(numbers[0]);
7      double sum = 0;
8
9      cout << "The numbers are: ";
10     for (const double& n : numbers)
11     {
12         cout << n << " ";
13         sum += n;
14     }
15     cout << "\nSum = " << sum << endl;
16     cout << "Size = " << size << endl;
17
18     double average = sum / size;
19     cout << "Average = " << average << endl;
20 }
```

```
The numbers are: 7.19 5.26 6.33 12.01 35.15 27.71
Sum = 93.65
Size = 6
Average = 15.6083
```

Приклад 7. Max/Min elements.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main() {
5      int elements[] = { -7, -20, 0, 5, -30, 50, 20 };
6      const size_t size = sizeof(elements) / sizeof(elements[0]);
7
8      // Initialize indexes of max and min elements
9      size_t maxIndex = 0;
10     size_t minIndex = 0;
11
12     // Initialize max and min elements
13     int maxElement = elements[maxIndex];
14     int minElement = elements[minIndex];
15
16     // Find max and min elements
17     for (size_t i = 1; i < size; ++i) {
18         if (elements[i] > maxElement) {
19             maxElement = elements[i]; // Update max if current element is greater
20             maxIndex = i;
21         }
22         if (elements[i] < minElement) {
23             minElement = elements[i]; // Update min if current element is smaller
24             minIndex = i;
25         }
26     }
27
28     // Output the max and min elements
29     std::cout << "Max element: elements[" << maxIndex << "] = " << maxElement << std::endl;
30     std::cout << "Min element: elements[" << minIndex << "] = " << minElement << std::endl;
31
32     return 0;
33 }
```

```
Max element: elements[5] = 50
Min element: elements[4] = -30
```


Багатовимірні масиви



FACULTY OF APPLIED
MATHEMATICS AND
INFORMATICS
LVIV UNIVERSITY

Багатовимірний масив (Двовимірний)

Двовимірний (матриця)
`int x[2][3];`



	Col 1	Col 2	Col 3	Col 4
Row 1	<code>x[0][0]</code>	<code>x[0][1]</code>	<code>x[0][2]</code>	<code>x[0][3]</code>
Row 2	<code>x[1][0]</code>	<code>x[1][1]</code>	<code>x[1][2]</code>	<code>x[1][3]</code>
Row 3	<code>x[2][0]</code>	<code>x[2][1]</code>	<code>x[2][2]</code>	<code>x[2][3]</code>

Тривимірний
`int x[2][3][4];`

Багатовимірний масив (Двовимірний)

```
typedef unsigned int size_t;           // стандартний тип для
                                        // розміру
```

```
const size_t rows=2;
const size_t columns=4;
double b[rows][columns]={10,20,30,40,50,60,70,80};
cout << b[0][0]+b[1][4];
```

	b[0][0]	b[0][1]	b[0][2]	b[0][3]	b[1][0]	b[1][1]	b[1][2]	b[1][3]
b	10	20	30	40	50	60	70	80

Насправді елементи багатовимірних масивів в пам'яті розміщені послідовно

Приклади

Приклад 6. Двовимірний масив.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      const size_t row = 3;
8      const size_t col = 3;
9      double arr[row][col] = { {10,20,30},{40,50,60},{70,80,90} }; // { 10,20,30,40,50,60,70,80,90 }
10     for (size_t i = 0; i < row; ++i)
11     {
12         for (size_t j = 0; j < col; ++j)
13         {
14             cout << arr[i][j] << ' ';
15         }
16         cout << endl;
17     }
```

10	20	30
40	50	60
70	80	90

Приклад 7. Сума макс. елементів рядків.

```
5 int main()
6 {
7     const size_t ROWS = 3;
8     const size_t COLS = 3;
9     int matrix[ROWS][COLS]; // Static 2D array for the matrix
10
11     // Input matrix elements
12     std::cout << "Enter the elements of the matrix [3x3]:\n";
13     for (int i = 0; i < ROWS; ++i) {
14         for (int j = 0; j < COLS; ++j) {
15             std::cout << "Element [" << i << "][" << j << "]: ";
16             std::cin >> matrix[i][j];
17         }
18     }
19     int sum = 0;
20     // Find and display the maximum element for each row
21     for (int i = 0; i < ROWS; ++i) {
22         int maxElement = matrix[i][0]; // Initialize max for the current row
23         for (int j = 1; j < COLS; ++j) {
24             if (matrix[i][j] > maxElement) {
25                 maxElement = matrix[i][j]; // Update max if current element is greater
26             }
27         }
28         std::cout << "Maximum element in row " << i << ": " << maxElement << std::endl;
29         sum += maxElement;
30     }
31     std::cout << "Sum of max rows elements " << sum << std::endl;
32
33     return 0;
34 }
```

```
Enter the elements of the matrix [3x3]:
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [1][0]: 4
Element [1][1]: 5
Element [1][2]: 6
Element [2][0]: 7
Element [2][1]: 8
Element [2][2]: 9
Maximum element in row 0: 3
Maximum element in row 1: 6
Maximum element in row 2: 9
Sum of max rows elements 18
```

Приклад 8. Тривимірний масив.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     char letters[2][2][2] = {
7         {
8             { 'A', 'B' },
9             { 'C', 'D' }
10        },
11        {
12            { 'E', 'F' },
13            { 'G', 'H' }
14        }
15    };
16
17    for (size_t i = 0; i < 2; ++i) {
18        for (size_t j = 0; j < 2; ++j) {
19            for (size_t k = 0; k < 2; ++k) {
20                cout << letters[i][j][k] << ' ';
21            }
22            cout << endl;
23        }
24        cout << endl;
25    }
26    return 0;
27 }
```

A	B
C	D
E	F
G	H

Дякую!



FACULTY OF APPLIED
MATHEMATICS AND
INFORMATICS
LVIV UNIVERSITY