**Тема 15. Структури даних**

15.1.Прості і складені типи даних

15.2. Масиви. Типові алгоритми обробки масивів

15.3 Стеки. Черги. Списки. Галузі застосування структур даних при програмуванні мультимедійних додатків

**Масив** – це набір елементів одного і того ж типу, об'єднаних загальним ім'ям.

**Одновимірний масив** ─ це фіксована кількість елементів одного й того ж типу, об'єднаних загальним ім'ям, де кожен елемент має свій номер.

Нумерація елементів масиву, як правило (наприклад в С#), починається з нуля, тобто, якщо масив складається з 10 елементів, то його елементи матимуть такі номери: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Одновимірний масив у сучасних мовах (С#) реалізується як об'єкт, тому його створення є двоступеневим процесом. Спочатку оголошується посилальна змінна на масив, потім виділяється пам'ять під необхідну кількість елементів базового типу, і посилальній змінній присвоюється адреса нульового елемента в масиві.

**Базовий тип** визначає тип даних кожного елемента масиву. Кількість елементів, які будуть зберігатися в масиві, визначається розмір масиву.

**Оголошення масиву й присвоювання йому значення**

У загальному випадку процес оголошення змінної типу масив, і виділення необхідного обсягу пам'яті може бути розділене. Крім того, на етапі оголошення масиву можна провести його ініціалізацію. Тому для оголошення одновимірного масиву може використовуватися одна з наведених долі форм запису.

**Форма 1.**

базовий\_тіп [ ] ім’я\_\_масива;

Наприклад:

int [ ] a;

Описано посилання на одновимірний масив, яке в подальшому може бути використане для:

адресації на вже наявний масив;

передачі масиву в метод як параметр відстроченого виділення пам'яті під елементи масиву.

**Форма 2.**

базовий\_тіп [ ] ім’я\_\_масива = new базовий\_тип [розмір];

Наприклад:

int [ ] a = new int [10];

Оголошено одновимірний масив заданого типу та виділена пам'ять під одновимірний масив зазначеного розміру. Адреса цієї області пам'яті записана в посилальну змінну. Елементи масиву дорівнюють нулю.

**Форма 3.**

базовий\_тип [ ] ім’я\_\_масива = {список ініціалізації};

Наприклад:

int [ ] a = {0, 1, 2, 3};

Виділена пам'ять під одновимірний масив, розмірність якого відповідає кількості елементів у списку ініціалізації. Адреса цієї пам'яті записана в посилальну змінну. Значення елементів масиву відповідає списку ініціалізації.

Звернення до елементів масиву відбувається за допомогою індексу, для цього потрібно вказати ім'я масиву та в квадратних дужках його номер. Наприклад, a [0], b [10], c [i].

Оскільки масив є набором елементів, об'єднаних загальним ім'ям, то обробка масиву зазвичай проводиться в циклі (див. результат виконання лабораторної роботи №4).

**Русский вариант общих положений**

**(для иностранных студентов)**

## Массивы

Массивы (например, в языке С#) это именованная совокупность однотипных данных. Доступ к элементам массива осуществляется с помощью индексов. Различают одномерные (вектора), двухмерные (матрицы) и n – мерные массивы.

### *Одномерные массивы*

Массив (например, в языке С#) объявляется следующим образом

тип [ ] имя массива;

int [ ] Vec;

int [ ] Vec = {5; -4; 75};

Далее создается массив

Имя массива = new тип элементов [количество элементов];

Vec = new int [3];

Объявление и создание массива можно объединить.

Тип [ ] имя массива = new тип[количество элементов];

int [ ] Vec = new int [3];

Нумерация элементов массива , как правило (например, в языке С#), идет с нуля. Таким образом, в примере начальный элемент массива – это Vec[0], a последний – Vec[2].

### *Двухмерные массивы*

Двухмерный массив объявляется с использованием одной разделительной запятой внутри квадратных скобок

int [,] Matr = new int [2,3];

Матрица состоит из 2 строк и 3 столбцов. В приведенном примере массив состоит из 6 элементов (Matr[0,0] – первый, Matr[1,2] – последний).

Трехмерный массив может быть создан следующим образом

int [,,] Kub = new int [3,3,3];

Инициализация двухмерного массива выглядит следующим образом

int [,] Matr = {(2,-2), (3,-22), (0,4)};