Едномерни масиви. Основни задачи. Увод в алгоритмите за сортиране. Библиотечна функция `sort`.

Моника Велчева

Мотивация — защо ни е нужен *масив*?

Имаме много свързани стойности (например оценки на учениците).

- С отделни променливи: `o1, o2, o3, o4, o5` — неудобно.
- **Масив** = една променлива, която съхранява много стойности.
- Подреждаме ги последователно и им даваме **индекси**.

Как изглежда масивът? (схема)

*Индексите започват от **0**. Последният индекс е **n-1**.*

Деклариране и достъп до елементи

```
int ocenka[5] = {5, 6, 5, 4, 6};
cout << ocenka[0]; // 5
cout << ocenka[4]; // 6
```

Обхождане на масив

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

Сбор и средно аритметично

```
int suma = 0;
for (int i = 0; i < 5; i++) suma += ocenka[i];
float sreden = (float)suma / 5;
cout << "Сбор: " << suma << ", Средна оценка: " << sreden << "\n";
```

Минимум и максимум

```
int minv = ocenka[0], maxv = ocenka[0];
for (int i = 1; i < 5; i++) {
    if (ocenka[i] < minv) minv = ocenka[i];
    if (ocenka[i] > maxv) maxv = ocenka[i];
}
cout << "Минимум: " << minv << ", Максимум: " << maxv << "\n";
```

Какво е сортиране? (схема)

Преди:

Подреждаме числата във възходящ ред.

Bubble Sort — примерен код

```
void bubble_sort(vector<int>& a) {
    int n = (int)a.size();
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - 1; j++) {
            if (a[j] > a[j + 1]) swap(a[j], a[j + 1]);
        }
    }
}
```

Библиотечна функция `sort`

- `sort(a.begin(), a.end())` — възходящ ред.
- `sort(a.rbegin(), a.rend())` — низходящ ред.
- Работи за числа, низове и структури.
- Използва бърз алгоритъм ($O(n \log n)$).

Пример — сортиране на структури

```
struct Uchenik { string ime; int ocenka; };
vector<Uchenik> v = {{"Ива",6}, {"Ана",5}, {"Марио",6}};

sort(v.begin(), v.end(), [](auto &a, auto &b){
    if (a.ocenka != b.ocenka) return a.ocenka > b.ocenka;
    return a.ime < b.ime;
});
```

Упражнения (в клас)

- Намерете броя на шестиците.
- Разменете местата на минималния и максималния елемент.
- Проверете дали масивът е сортиран.
- Изведете елементите в обратен ред.

Допълнителни задачи (за домашно)

- Бройте колко пъти се среща всяко число.
- Намерете броя на различните елементи.
- Намерете втория по големина елемент.
- Сортирайте низове по дължина.
- Подредете ученици по име и оценка.

Обобщение

- Масив — последователност от стойности.
- Индекси започват от 0.
- Основни операции: обхождане, сбор, средно, макс/мин.
- Сортиране: подреждаме данните — Bubble Sort и `sort()`.