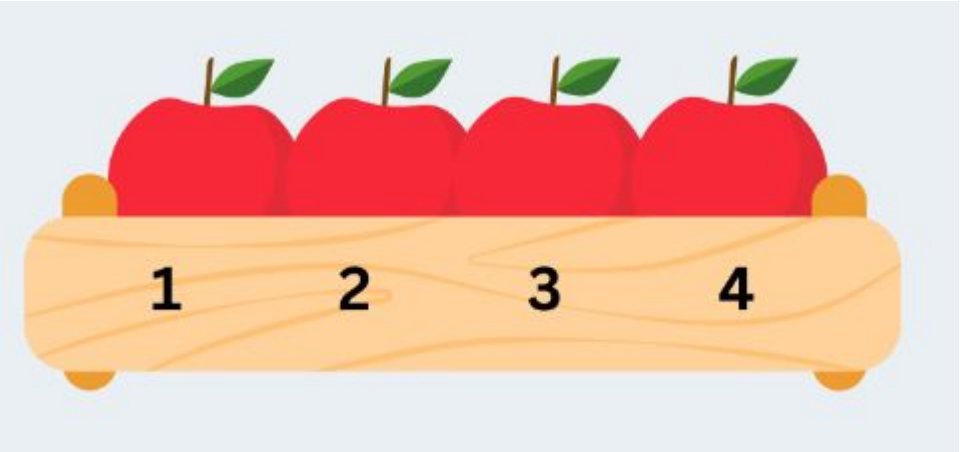
05. Масиви

УП практикум, 2ра група
Виолета Кастрева
Богомил Стоянов

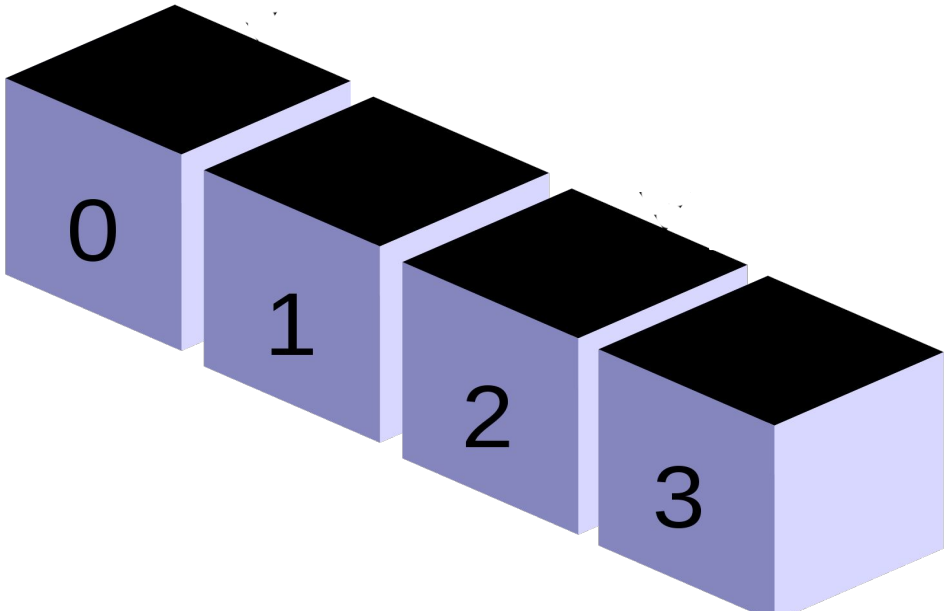
Какво е масив?

Представете си, че имате рафт и на него имате няколко отделения, в които съхранявате предмети. Сега помислете, че всяко отделение може да съхранява само един артикул и всички артикули са от един и същи тип, като книги или обувки. В програмирането този рафт може да се разглежда като масив и всяко отделение е индекс на масива.



Е, защо ни е тая работа?

Без масиви, ако искате да съхраните 10 числа, ще ви трябват 10 различни променливи. С масив се нуждаете само от една "групираща" променлива. Той опростява съхранението и достъпа до поредица от елементи от същия тип.



Как да декларираме и инициализираме масиви?

За да декларирате масив, трябва да посочите неговия тип, име и размер.

```
1 int numbers[10];
2 // Declares an array named 'numbers'
3 //that can store 10 integers.
4
5
```

Инициализация

Можете да зададете стойности на масив по време на декларация:

```
1 int days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
2 // Array storing days in each month.
3
4
```

Достъп до елементите

Имате достъп до всеки елемент в масива, като използвате неговия индекс. Индексите започват от 0 и достигат до единица по-малка от размера на масива.

За нашия набор от дни:

days[0] се отнася за 31 (януари)

days[1] се отнасят за 28 или 29 (февруари, в зависимост дали е високосна година)

... и така нататък.

Итериране през масиви (запомнете я тази гадна думичка)

Циклите обикновено се използват с масиви. Например, може да искате да отпечатате всички елементи на масив:

```
1 for(int i = 0; i < 12; i++) {
2     cout << days[i] << endl;
3 }
4
```

Внимавайте с индексите

Не искаме да ни гърми кода, това ще хвърли грешка

```
1 int numbers[5];
2 cout << numbers[10] << endl;
3 // Accessing beyond the array's size!
4
5
```

Край

Задачите са по-гадни от увода