

Theory: Fixed-size array

🕒 15 minutes 0 / 5 problems solved

Skip this topic

Start practicing

3428 users solved this topic. Latest completion was 19 minutes ago.

In programs, data is represented in the format of numbers, characters, strings, or other more complex objects. Often, some of these objects can be naturally grouped together.

For example, assume that you conducted an experiment and got measurements for some process. The measurements might correspond to temperature, distance, time, or something else. In such a case, it would be convenient not to store each measurement as a separate variable but to process all of them together as a single unit.

To efficiently deal with such cases, most programming languages provide a structure called a **fixed-size array**. The structure allows us to sequentially store a collection of elements of the same type and process them as a single unit. In this topic, we will consider this structure in more detail and learn some basic operations used for arrays.

§1. A fixed-size array

A fixed-size array is a structure for storing a collection of elements of the same type. As you can conclude from the title, the size of an array is a constant: once an array is created, its size cannot be changed. The elements of a fixed-sized array are stored sequentially. Each element has an index, that corresponds to its sequence number in the array. The first element has the index **0**, the second element has the index **1**, the last element has the index $n - 1$, where n is the total number of elements in the array. Given below is an example of a fixed-size array that stores five floating-point numbers:

Index	0	1	2	3	4
Element	10.8	14.3	13.5	12.1	9.7

The first element of the array is **10.8** and its index is **0**, the second one is **14.3** with the index **1**. The last element is **9.7** with the index **4**. Using pseudocode, the array can be represented as follows:

```
1 | measurements = [10.8, 14.3, 13.5, 12.1, 9.7]
```

A variable named *measurements* unites the numbers to a single unit.

§2. Getting the accessing elements

Programming languages provide a set of standard methods for arrays processing. There is one of them used most frequently. It is a method for accessing an element by its **index**:

```
1 | measurements = [10.8, 14.3, 13.5, 12.1, 9.7]
2 | value = measurements[2] # value is equal to 13.5
```

We can not only **read** but also **modify** elements of the array:

```
1 | measurements[2] = 3.7
```

Now, the array looks like this:

```
1 | measurements = [10.8, 14.3, 3.7, 12.1, 9.7]
```

Current topic:

[Fixed-size array](#) ...

Topic depends on:

✓ [Data structures](#) ...

Topic is required for:

[Dynamic array](#) ...

Table of contents:

[1 Fixed-size array](#)

[§1. A fixed-size array](#)

[§2. Getting the accessing elements](#)

[§3. Calculating the mean of array elements](#)

[§4. Summary](#)

[Feedback & Comments](#)

Along with the two, there are some other more sophisticated methods for arrays processing such as sorting, reversing, methods for searching elements, and some others. When you use a particular programming language, check the documentation of the standard library to see what methods the language provides.

§3. Calculating the mean of array elements

Let's consider a simple example of how arrays can be processed. Given an array of numbers, our task will be to calculate the mean of elements in the array. Mean of array elements is a number equal to the sum of all array elements divided by their number. We will consider how it can be done for our array of measurements:

```
1 | measurements = [10.8, 14.3, 13.5, 12.1, 9.7]
```

The procedure is the following:

```
1 | sum = 0
2 |
3 | for (i = 0; i < len(measurements); i = i + 1):
4 |     sum += measurements[i]
5 |
6 | mean = sum / len(measurements) # 12.08
```

Here, we initialize the variable `sum` with `0`, then, we start iterating through the range from `0` to `4` accessing each element of an array by its index and add it to the total sum. After that, we divide the obtained sum by the length of the array getting the mean value for the elements. The length of the array `measurements` is known in advance.

§4. Summary

A fixed-size array is a structure that allows us to sequentially store a collection of elements of the same type. The most frequently used method for arrays is accessing an element by its index.

To get information about other methods, check the standard library of a programming language you use. Use a fixed-size array if you need to process a collection of objects of similar type and you know the number of objects in advance.

 Report a typo

265 users liked this theory. 28 didn't like it. What about you?



Start practicing

[Comments \(20\)](#)[Hints \(0\)](#)[Useful links \(1\)](#)[Show discussion](#)