

Java → Exception handling → [Array exceptions](#)

1801 users solved this topic. Latest completion was about 6 hours ago.

# Theory: Array exceptions

🕒 10 minutes 0 / 5 problems solved

Skip this topic

Start practicing

No wonder that different types of exceptions may occur when your program processes an array. To avoid them, you should be aware of the situations where you are at risk of having one and stick to a set of commonly used practices. Now let's learn what exactly we are dealing with here.

## §1. NullPointerException

As you probably know by now, an array is a reference type, which means its variable can be `null`, and that may lead to NPE.

```
1 int[] numbers = null;
2 int size = numbers.length; // It throws NPE
```

We will not dwell on this since we suppose that you are already familiar with NPE and how to avoid it by using additional checks in your code:

```
1 int size = numbers == null ? 0 : numbers.length;
```

## §2. NegativeArraySizeException

If you try to create an array with a negative size, your code will compile successfully, but this line will throw `NegativeArraySizeException` while executing.

```
1 int negSize = -1;
2 int[] numbers = new int[negSize]; // an exception here
```

It's not very likely that you'll face this exception as a developer, but it makes sense to keep it in mind. To avoid it, simply do not use variables that can have a negative size when setting the size of an array.

An array may have a size of zero or one. If this is the case, the code will be successfully compiled.

## §3. ArrayIndexOutOfBoundsException

This is a fairly common exception that occurs while working with arrays. It is caused by attempting to access a non-existent element of the array.

```
1 int[] array = { 1, 2, 3 }; // an array of ints
2
3 int n1 = array[2]; // n1 is 3
4 int n2 = array[3]; // Exception
```

In this code, the last line produces `ArrayIndexOutOfBoundsException` since the last index of the array in question is 2.

The code will throw the same exception if we try to access an element with a negative index:

```
1 array[0]; // OK
2 array[-1]; // Exception
```

Since a string can be considered as a sequence of characters, a similar exception may occur when accessing a non-existing element of a string. It is called `StringIndexOutOfBoundsException`.

To avoid the `ArrayIndexOutOfBoundsException`, we may check if the given index belongs to the interval `[0, length - 1]`.

Current topic:

[Array exceptions](#) ...

Topic depends on:

✓

[Array](#) ...

Stage 2

✓

[NPE](#) ...

Table of contents:

[↑ Array exceptions](#)[§1. NullPointerException](#)[§2. NegativeArraySizeException](#)[§3. ArrayIndexOutOfBoundsException](#)[Feedback & Comments](#)

For example, let's take a look at a program, displaying an element of the array by the index, provided in the input. If the index is out of bounds, the program prints a message instead of throwing an exception.

```
1 public class NoIndexOutOfBoundsExceptions {
2
3     public static void main(String[] args) {
4         int[] hardCodedArray = { 3, 2, 4, 5, 1 };
5
6         Scanner scanner = new Scanner(System.in);
7
8         int index = scanner.nextInt();
9
10
11         if (index < 0 || index > hardCodedArray.length - 1) {
12
13             System.out.println("The index is out of bounds.");
14
15         } else {
16
17             System.out.println(hardCodedArray[index]);
18
19         }
20     }
21 }
```

Here are some possible inputs and the corresponding outputs of the program:

- the index is 0, the program outputs "3";
- the index is 1, the program outputs "2";
- the index is 4, the program outputs "1";
- the index is -1, the program outputs "The index is out of bounds.";
- the index is 5, the program also outputs "The index is out of bounds.".

That is how we can avoid `ArrayIndexOutOfBoundsException`s by using a conditional statement and the `length` property of an array.

 Report a typo

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



Start practicing

[Comments \(1\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)