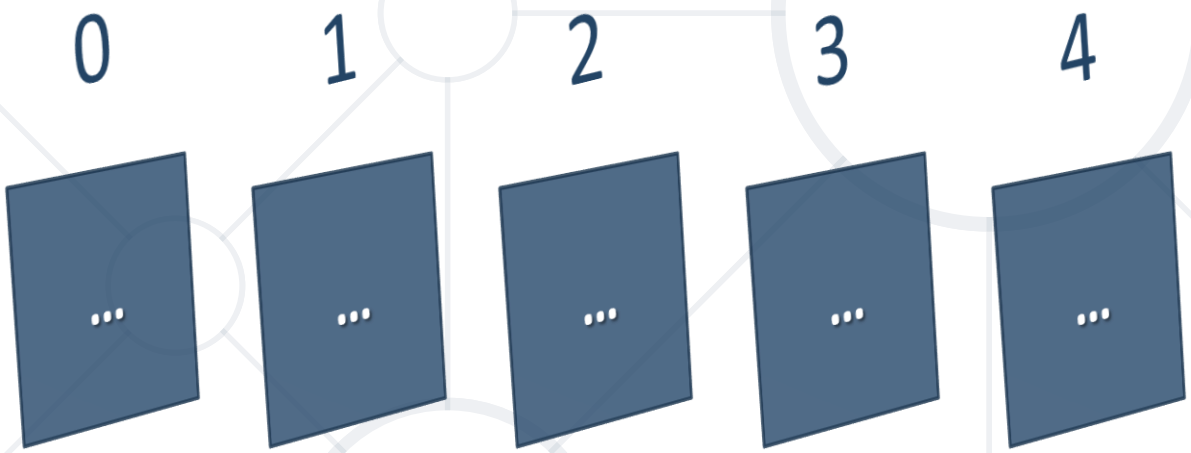
Nested Loops



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

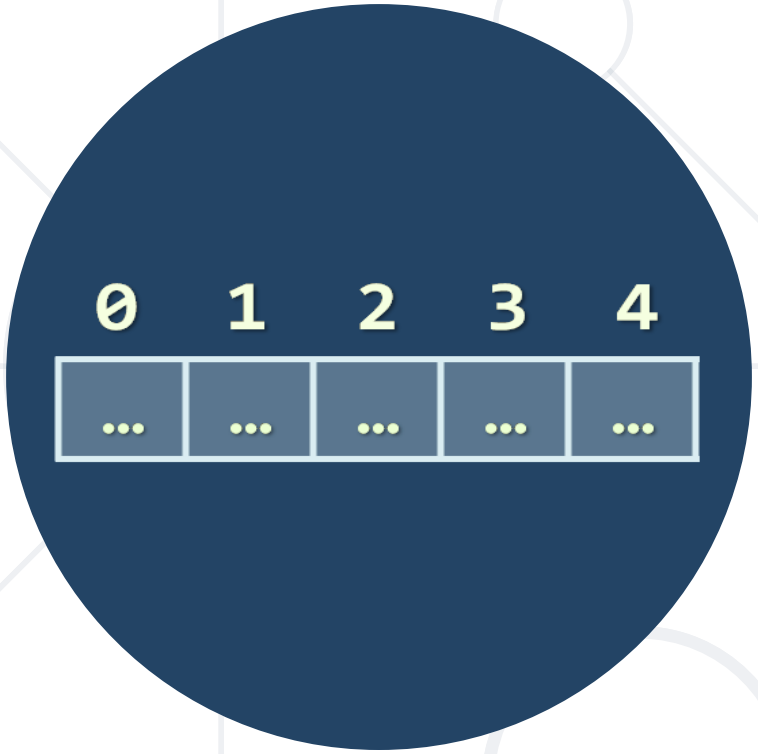
Have a Question?

Software University

sli.do

#prgm-for-qa

Table of Contents



Arrays

Working with Arrays of Elements

What are Arrays?

- In programming, an **array** is a **sequence of elements**

- Arrays have **fixed size** (**Array.Length**) cannot be resized
- Elements are of the **same type** (e. g. integers)
- Elements are numbered from **0** to **length-1**

Working with Arrays

- **Allocating** an array of 10 integers:

```
int[] numbers = new int[10];
```

All elements are initially == 0

- **Assigning values** to the array elements:

```
for (int i = 0; i < numbers.Length; i++)
  numbers[i] = 1;
```

The length holds the number of array elements

- **Accessing** array elements by index:

```
numbers[5] = numbers[2] + numbers[7];
numbers[10] = 1; // IndexOutOfRangeException
```

The [] operator accesses elements by index

Days of Week – Example

- The days of a week can be stored in an **array of strings**:

```
string[] days = {
   "Monday",
   "Tuesday",
   "Wednesday",
   "Thursday",
   "Friday",
   "Saturday",
   "Sunday"
};
```

Operator	Value
days[0]	Monday
days[1]	Tuesday
days[2]	Wednesday
days[3]	Thursday
days[4]	Friday
days[5]	Saturday
days[6]	Sunday

Problem: Day of Week

- Enter a **day number** [1…7]
- Print the **day name** or "**Invalid day!**"

```
string[] days = {"Monday", "Tuesday", "Wednesday", "
Thursday", "Friday", "Saturday", "Sunday"};
int day = int.Parse(Console.ReadLine());
if (day >= 1 && day <= 7)
  Console.WriteLine(days[day - 1]);
else
 Console.WriteLine("Invalid day!");
```

The first day in our array is on index 0, not 1.

Reading Arrays

Reading Arrays From the Console

- First, read the array **length** from the console :

```
int length = int.Parse(Console.ReadLine());
```

- Next, create an array of given size **n** and read its **elements**:

```
int[] arr = new int[length];
for (int i = 0; i < length; i++)
{
  arr[i] = int.Parse(Console.ReadLine());
}
```

Reading Array Values from a Single Line

- Arrays can be read from a **single line** of **space separated values**

```
2 8 30 25 40 72 -2 44 56
```

```
string values = Console.ReadLine();
string[] items = values.Split(" ");
int[] arr = new int[items.Length];

for (int i = 0; i < items.Length; i++)
  arr[i] = int.Parse(items[i]);
```

Shorter: Reading Array from a Single Line

- Read an array of integers using functional programming:

```
string inputLine = Console.ReadLine();
string[] items = inputLine.Split(" ");
int[] arr = items.Select(int.Parse).ToArray();
```

```
int[] arr = Console.ReadLine().Split(", ")
                .Select(int.Parse).ToArray();
```

Problem: Sum an Array

- Read an **array of integers** (from a single line)
- Print the **sum** of all items

`5 3 6 3 4`	→	`21`
`20 30 -5`	→	`45`
`25 45 -3`	→	`67`

Solution: Sum an Array

```
int[] numbers =     Console.ReadLine()
                    .Split(" ")
                    .Select(int.Parse)
                    .ToArray();
int sum = 0;

for (int i = 0; i < numbers.Length; i++)
{
    sum += numbers[i];
}

Console.WriteLine(sum);
```

Printing Arrays

Printing Arrays to the Console

- To print all array elements, a **for**-loop can be used
  - Separate elements with white space or a new line

```
string[] arr = {"one", "two"};
// Process all array elements
for (int i = 0; i < arr.Length; i++)
{
    Console.WriteLine("arr[{i}] = {arr[i]}");
}
```

Printing Arrays with string.Join(…)

- Use `string.Join(separator, array)` to print an array:

```
string[] strings = { "one", "two" };
Console.WriteLine(string.Join(" ", strings)); //one two
```

```
int[] arr = { 1, 2, 3 };
Console.WriteLine(string.Join(", ", arr)); //1, 2, 3
```

Problem: Reverse an Array

- Read an array of integers (**n** lines of integers), **reverse** it and print its elements on a single line, space-separated:

```
3
10
20
30
```

→

```
30 20 10
```

```
4
-1
20
99
5
```

→

```
5 99 20 -1
```

Solution: Reverse an Array

```
// Read the array (n lines of integers)
int n = int.Parse(Console.ReadLine());
int[] arr = new int[n];
for (int i = 0; i < n; i++)
  arr[i] = int.Parse(Console.ReadLine());
// Print the elements from the last to the first
for (int i = n - 1; i >= 0; i--)
  Console.Write(arr[i] + " ");
```

Foreach Loop

Foreach Loop

- **Iterates** through all elements in a collection
- Cannot access the current index
- **Read-only**

```
foreach (var item in collection)
{
    // Process the value here
}
```

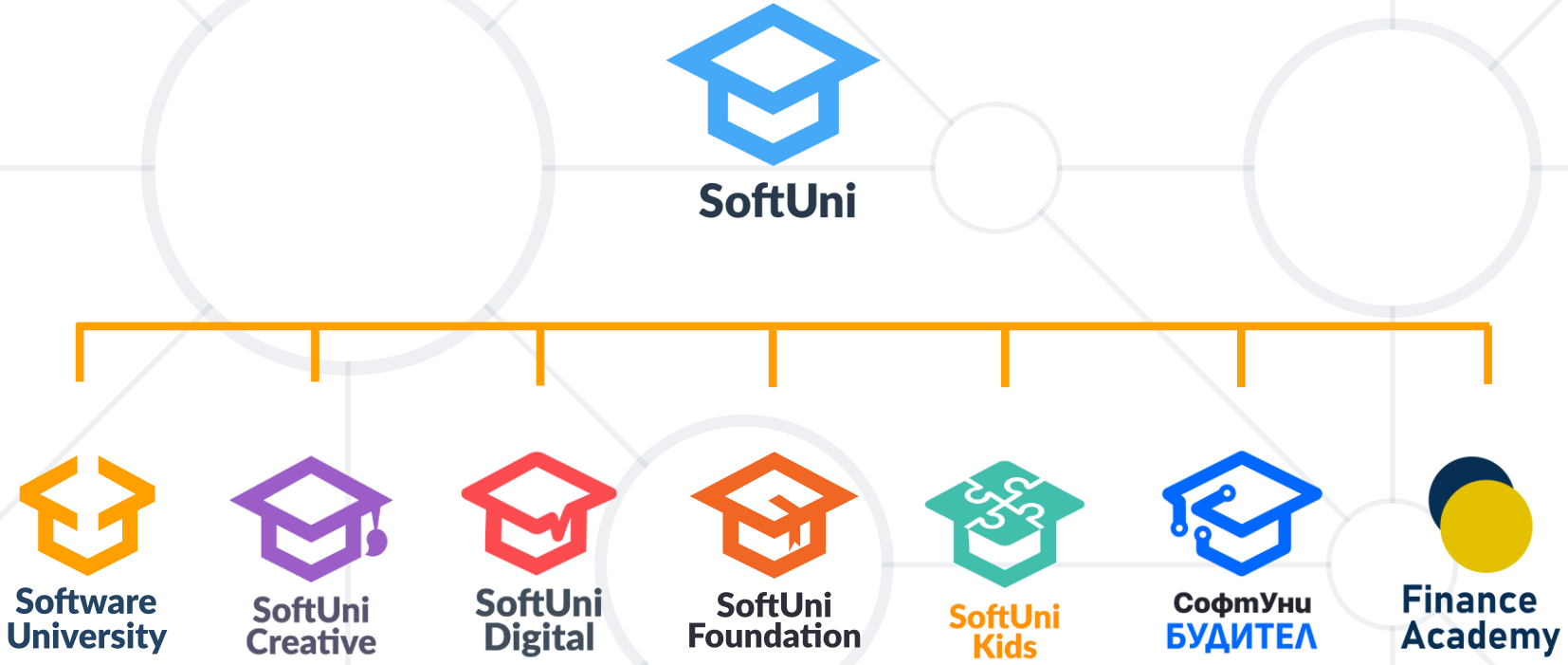
Print an Array with Foreach

```
int[] numbers = { 1, 2, 3, 4, 5 };
foreach (int number in numbers)
{
    Console.Write($"{number} ");
}
```

Summary

- **Arrays** hold a **sequence** of elements
  - Elements are numbered from **`0`** to **`length - 1`**
- **Creating** (allocating) an array
- **Reading** and **printing** arrays
- Accessing array elements by **index**

Questions?



SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank
Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR
Serving the high achievers

SOFTWARE GROUP

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg
- © Software University – https://softuni.bg