Functional Programming

Lambda Expressions

SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

Table of Contents

Have a Question?

SoftUni Foundation

sli.do

#csharp-advanced

Functional Programming

Paradigms, Concepts

What is Function?

- Mathematic function

Name **f(x) = x2**

Input Output

- A function is a special relationship where **each input** has a **single output**

X	f(x)
3	9
1	1
0	0
4	16
-4	16

Functional Programming

- Functional programming is declarative
- Its main focus is on "what to solve" and not "how to solve"
- Functions can be:
  - First-Class
  - Higher-Order – they either take other functions as arguments or return them as results

Functional Programming (2)

- Treats computation as the evaluation of mathematical functions, avoiding state and mutable data
- There can't be any information accessed beside the input variables
- The output value of a function depends only on the arguments that are passed to the function

Lambda Expressions

Implicit / Explicit Lambda Expressions

Lambda Expressions

- A lambda expression is an anonymous function containing expressions and statements
- Lambda syntax

```
(parameters) =>
{body}
```

  - Use the lambda operator "**=>**" (**goes to**)
  - Parameters can be enclosed in parentheses **( )**
  - The body holds the expression or statement and can be enclosed in braces **{}**

Lambda Expressions (2)

SoftUni Foundation

- Implicit lambda expression

```
msg =>
Console.WriteLine(msg);
```

- Explicit lambda expression

```
(String msg) =>
{ Console.WriteLine(msg); }
```

- Zero parameters

```
() =>
{ Console.WriteLine("hi"); }
```

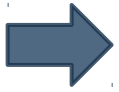
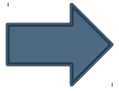
```
() =>
MyMethod();
```

- More parameters

```
(int x, int y) => { return x +
y; }
```

Problem: Sort Even Numbers

- Read integers from the console
- Print the **even numbers**, sorted in ascending order
- Use **two Lambda Expressions**
- Examples:

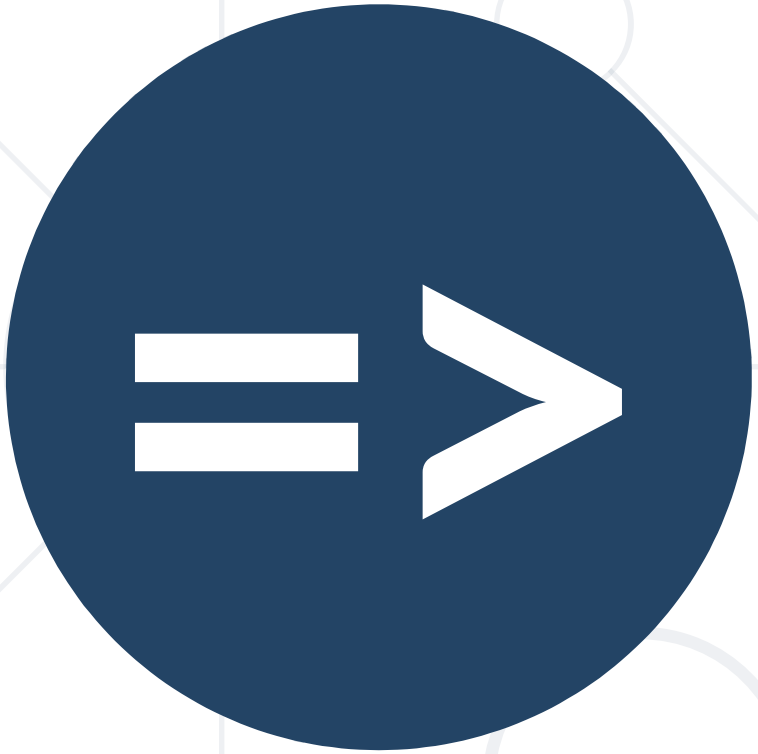
`4, 2, 1, 3, 5, 7, 1, 4, 2, 12`		`2, 2, 4, 4, 12`
`1, 3, 3, 4, 5, 6, 10, 9, 8, 2`		`2, 4, 6, 8, 10`
`1, 3, 4, 13, 10, 23, 45, 5, 1`		`4, 10`

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Solution: Sort Even Numbers

```
int[] numbers = Console.ReadLine()
   .Split(new string[] { ", " },
      StringSplitOptions.RemoveEmptyEntries)
         .Select(n => int.Parse(n))
         .Where(n => n % 2 == 0)
         .OrderBy(n => n)
         .ToArray();
string result = string.Join(", ", numbers);
Console.WriteLine(result);
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab



Delegates

Func<T, V>, Action<T>

Delegates

SoftUni Foundation

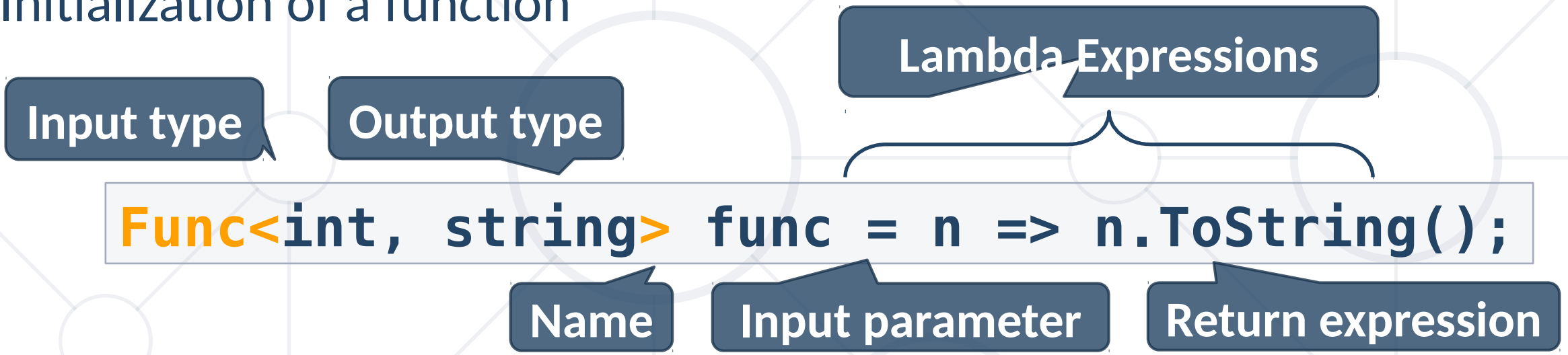
- A **delegate** is a type that represents references to methods with a particular parameter list and return type
- Used to pass **methods as arguments** to other methods
- Can be used to define callback methods

```
public delegate int Multiplier(int x, int
y);
Multiplier calc = (x, y) => x * y;
```

Generic Delegates - Func<T, V>

SoftUni Foundation

- Initialization of a function



- Input and output type can be **different types**
- Input and output type **must be from the declared type**
- **Func** generic delegate uses type parameters to define the number and types of input parameters and returns the type of the delegate

Generic Delegates - Action<T>

- In .NET `Action<T>` is a void method:

```
private void Print(string
message)
{  Console.WriteLine(message);
}
```

- Instead of writing the method we can do:

```
Action<string> print = message =>
    Console.WriteLine(message);
```

- Then we use it like that:

```
print("pesho");      //
pesho
print(5.ToString()); // 5
```

Problem: Sum Numbers

- Read numbers from the console
- Use your own **function to parse** each element
- Print the **count** of numbers
- Print the **sum**

```
4, 2, 1, 3, 5, 7, 1, 4, 2, 12
```

```
10
41
```

```
85, 47, 91, 32, 83, 75, 81, 2
```

```
8
496
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

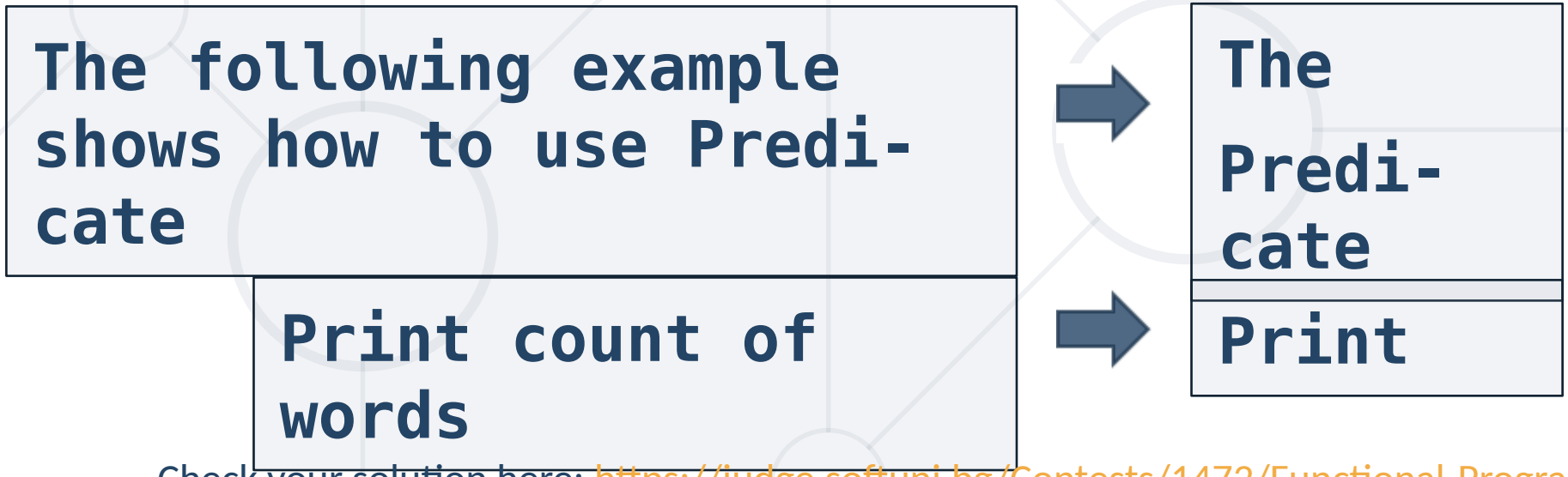
Solution: Sum Numbers

```
string input = Console.ReadLine();
Func<string, int> parser = n =>
int.Parse(n);
int[] numbers = input.Split(new string[] {",
"},
                StringSplitOptions.RemoveEmptyEn-
tries)
                .Select(parser).ToArray();
Console.WriteLine(numbers.Length);
Console.WriteLine(numbers.Sum());
```

- Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Problem: Count Uppercase Words

- Read a text from the console
- Filter only words, that **start** with a **capital** letter
- Use **`Predicate`**
- Print each of the words on a new line



Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Solution: Count Uppercase Words

SoftUni Foundation

```
Func<string, bool> checker = n => n[0] ==
n.ToUpper()[0];
var words = Console.ReadLine().Split(new string[] {"
"},
                StringSplitOptions.RemoveEmptyEntries)
                    .Where(checker)
                    .ToArray();
foreach (string word in words)
{
  Console.WriteLine(word);
}
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Problem: Add VAT

- Read from the console **prices of items**
- Add **VAT** of 20% to all of them
- Use **`UnaryOperator`**

```
1, 3, 5, 7
```

```
1.20
3.60
6.00
8.40
```

```
1, 3, 5, 7
```

```
1.20
3.60
6.00
8.40
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Solution: Add VAT

```
double[] prices = Console.ReadLine()
    .Split(new string[] { ", " },
        StringSplitOptions.RemoveEmptyEn-
tries)
    .Select(double.Parse)
    .Select(n => n * 1.2)
    .ToArray();
foreach (var price in prices)
  Console.WriteLine($"{price:f2}");
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Passing Functions to Method

- We can pass **Func<T>** to methods:

```
private int Operation(int number, Func<int, int>
operation)
{
   return operation(number);
}
```

- We can use the method like that:

```
int a = 5;
int b = Operation(a, number => number *
5); // 25
int c = Operation(a, number => number -
3); // 2
int d = Operation(b, number => number %
2); // 1
```

Problem: Filter by Age

- Read from the console **n people** with their **age**
- Read a **condition** and an age **filter**
- Read a format type for output and filter the people

```
4
Pesho, 20
Mimi, 29
Ico, 31
Simo, 16
older
20
name age
```

```
Pesho - 20
Mimi - 29
Ico - 31
```

```
4
Pesho, 20
Gosho, 18
Mimi, 29
Simo, 16
younger
20
name
```

```
Gosho
Simo
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Solution: Filter by Age

SoftUni Foundation

```
// TODO: Read data from the console
Func<int, bool> tester = CreateTester(condition,
age);
Action<KeyValuePair<string, int>> printer =

CreatePrinter(format);
PrintFilteredStudent(people, tester, printer);
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Solution: Filter by Age (2)

SoftUni Foundation

```
public static Func<int, bool> CreateTester
                (string condition, int age)
{
  switch (condition) {
    case "younger": return x => x < age;
    case "older": return x => x >= age;
    default: return null;
  }
}
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Solution: Filter by Age (3)

SoftUni Foundation

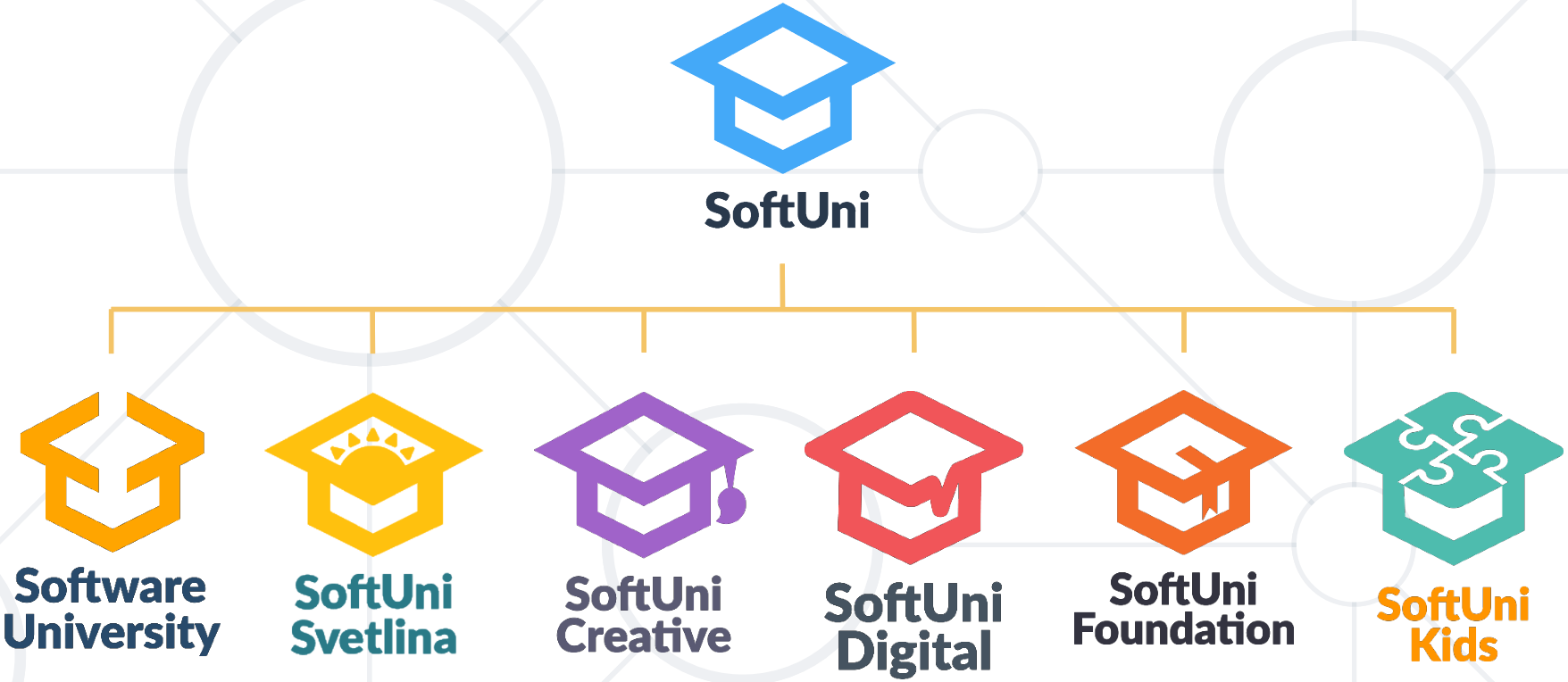
```
public static Action<KeyValuePair<string, int>>
                                    CreatePrinter(string
format)
{
  switch (format)
    case "name":
      return person =>
Console.WriteLine($"{person.Key}");
    // TODO: complete the other cases
    default: return null;
}
```

Check your solution here: https://judge.softuni.bg/Contests/1472/Functional-Programming-Lab

Summary

- Lambda expressions are **anonymous functions** used with delegates
- **Func<T, TResult>** is a function that returns TResult type
- **Action<T>** is a void function
- Functions can be submitted as method parameters

Questions?


SoftUni
Software University
SoftUni Svetlina
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS®

STEMO®
Computer Systems & Software

SoftUni Organizational Partners

SoftUni Foundation

Trainings @ Software University (SoftUni)

SoftUni Foundation

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

SoftUni Foundation

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license