GitHub This repository Search Explore Features Enterprise Pricing Sign up Sign in

TelerikAcademy / Object-Oriented-Programming Watch 17 Star 8 Fork 30

Branch: master

Object-Oriented-Programming / 03. Extension-Methods-Delegates-Lambda-LINQ / README.md

evlogihr on 13 Mar Problem 16. changed property in Student to be GroupNumber in order to…

2 contributors

81 lines (59 sloc) 3.84 KB Raw Blame History

Homework: Extension-Methods-Delegates-Lambda-LINQ

Problem 1. StringBuilder.Substring

- Implement an extension method `Substring(int index, int length)` for the class `StringBuilder` that returns new `StringBuilder` and has the same functionality as `Substring` in the class `String` .

Problem 2. IEnumerable extensions

- Implement a set of extension methods for `IEnumerable<T>` that implement the following group functions: sum, product, min, max, average.

Problem 3. First before last

- Write a method that from a given array of students finds all students whose first name is before its last name alphabetically.
- Use LINQ query operators.

Problem 4. Age range

- Write a LINQ query that finds the first name and last name of all students with age between 18 and 24.

Problem 5. Order students

- Using the extension methods `OrderBy()` and `ThenBy()` with lambda expressions sort the students by first name and last name in descending order.
- Rewrite the same with LINQ.

Problem 6. Divisible by 7 and 3

- Write a program that prints from given array of integers all numbers that are divisible by 7 and 3. Use the built-in extension methods and lambda expressions. Rewrite the same with LINQ.

Problem 7. Timer

- Using delegates write a class `Timer` that can execute certain method at each `t` seconds.

Problem 8.* Events

- Read in MSDN about the keyword `event` in C# and how to publish events.

- Re-implement the above using .NET events and following the best practices.

Problem 9. Student groups

- Create a class `Student` with properties `FirstName` , `LastName` , `FN` , `Tel` , `Email` , `Marks` (a List), `GroupNumber` .
- Create a `List<Student>` with sample students. Select only the students that are from group number 2.
- Use LINQ query. Order the students by FirstName.

Problem 10. Student groups extensions

- Implement the previous using the same query expressed with extension methods.

Problem 11. Extract students by email

- Extract all students that have email in `abv.bg` .
- Use `string` methods and LINQ.

Problem 12. Extract students by phone

- Extract all students with phones in Sofia.
- Use LINQ.

Problem 13. Extract students by marks

- Select all students that have at least one mark `Excellent` (`6`) into a new anonymous class that has properties – `FullName` and `Marks` .
- Use LINQ.

Problem 14. Extract students with two marks

- Write down a similar program that extracts the students with exactly two marks " `2` ".
- Use extension methods.

Problem 15. Extract marks

- Extract all `Marks` of the students that enrolled in 2006. (The students from 2006 have 06 as their 5-th and 6-th digit in the FN).

Problem 16.* Groups

- Create a class `Group` with properties `GroupNumber` and `DepartmentName` .
- Introduce a property `GroupNumber` in the `Student` class.
- Extract all students from "Mathematics" department.
- Use the `Join` operator.

Problem 17. Longest string

- Write a program to return the string with maximum length from an array of strings.
- Use LINQ.

Problem 18. Grouped by GroupNumber

- Create a program that extracts all students grouped by `GroupNumber` and then prints them to the console.
- Use LINQ.

Problem 19. Grouped by GroupName extensions

- Rewrite the previous using extension methods.

Problem 20.* Infinite convergent series

- By using delegates develop an universal static method to calculate the sum of infinite convergent series with given precision depending on a function of its term. By using proper functions for the term calculate with a 2-digit precision the sum of the infinite series:
  - 1 + 1/2 + 1/4 + 1/8 + 1/16 + …
  - 1 + 1/2! + 1/3! + 1/4! + 1/5! + …
  - 1 + 1/2 - 1/4 + 1/8 - 1/16 + …