UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

SP from Algorithms and Data Structures 1 - Fourth level

Assignment

Using the algorithm from the first level and using a user-specified predicate (from the first level), obtain a sequential direct data structure from the selected node of the hierarchy (you can navigate to the node using the solution from the second level).

Program a universal sorting algorithm to sort the data from the data structure, based on the comparator[1]. We will make use of comparators that will compare:

- `compareAlphabetical`: the name of the first territorial unit is alphabetically preceded by the name of the second territorial unit.
- `comparePopulation`: compares whether the population of a given sex in a given year of the first territorial unit is smaller than the population of the same sex in the same year of the second territorial unit, based on a given year (2020-2024) and a given sex (male, female, total).

After sorting, print out the sorted data along with the values on which they were sorted.

Tips

- Encapsulate the sorting algorithm in a separate object.
- Design comparators (in our case `compareAlphabetical` and `comparePopulation`) as lambda functions, functional objects or virtual methods (the first 2 options are preferred).

Evaluation

Demonstration of functionality (optional)

- There is none.

During defense

- You must implement level three functionality (score at least a 1p for level three) to get a level four score.
- Points are not dependent on bonus level.
- The sorting algorithm is in a dedicated separate object (i.e. it is not in a non-member method or a member method of another object that calls it) - max 5pts.
- It is possible to change the comparator in the sorting algorithm without modifying the code of the algorithm itself (using virtual method max. 3pts, lambda function max. 5pts or function object max. 5pts).
- Comparer `compareAlphabetical` works with diacritics (accents) – max. 5pts, without diacritics max. 3pts.
- Comparer `comparePopulation` – max. 5pts.

Documentation

The documentation is submitted with the final version of the SP at the end of the semester. The documentation must be prepared according to the published requirements (document SP rules).

[1] A comparator is a function that takes two compared objects and returns -1 if the value of the compared property of the first object is less than the value of the property of the second object; 0 if the values are the same; and +1 if the value of the compared property of the first object is greater than the value of the property of the second object.