Mandatory Activity. Functional Paradigm. Lab 05.

Activity

You are given C# code files that provide services to create random instances of `Employee` and `Videogame` classes to facilitate testing. Using this code, implement the following generic higher-order functions in a different project.

- `Find`: Returns the first element in a collection that fulfils a specific predicate. If no suitable element exists, the default value is returned.
    - Test it to search for employees by name and by NIF numbers ending in a given letter. Test to search for names that are not possible for employees.
    - Test it to search PS3 games, and for games with a non-existing Genre (like "*MusicalRPG*").

- `Filter`: Returns all the elements in a collection that fulfils a given predicate.
    - Test the same scenarios described for `Find`.

- `Reduce`: Returns the application of a function to all the elements of a collection, so **a single value** is returned storing the computation done with all the collection elements. This generic method uses two generic types: type of the elements in the collection and type of the returned value. For example, `Reduce` can be used to compute the summation of all the sales of a `Videogame` collection. The first type is the type of the elements in the collection (`Videogame`). The second one is the type of the result (a summation, `double`). In our example, the function passed as a parameter to `Reduce` receives a `double` and a `Videogame` parameter, and returns a `double`. On each iteration, the first `double` parameter has the summation of the sales so far, and the function must return these number plus the sales of the currently processed `Videogame`.

    - Test it to compute the summation of the Europe sales of the videogames.
    - **Note**: To perform the following tests, an optional parameter must be added with the initial value of the value that accumulate the results.
        - Test it to compute the summation of the sales of the videogames, separated by region (10123500 units on Europe, 3400.657 on Japan, etc.)
        - Test it to group employees by name; e.g., 10 employees named "María", 6 employees named "Pedro"…)

Test its correct behaviour by using the testing tool of Visual Studio.

Use the appropriate programming language features learned so far.