Mandatory Activity. Functional Paradigm. Lab 05.

This activity must be autonomously done by the student. **It must be done prior to the following laboratory class**. It will be used as part of the following laboratory.

Activity

The Angle.cs and Person.cs files implement the `Angle` and `Person` class. Factory.cs provides services to create instances of `Angle` and `Person`, useful for testing purposes. Using this code, implement the following higher-order functions.

1. `Find`: Returns the first element in a collection that fulfills a specific predicate. If no element equal to the parameter exists, a default value must be returned.
   - Test it to search for people by name and by id numbers ending in a given letter.
   - Test it to search for right angles, and angles in a given quadrant.
2. `Filter`: Returns all the elements in a collection that fulfills a given predicate.
   - Test the same scenarios described for `Find`.
3. `Reduce`: A function is applied to all the elements in a collection, returning a single value. This generic method has two generic arguments. For example, `Reduce` can be used to compute the summation of all the degrees of an `Angle` collection. The first type is the type of the reduced value (`double`) and should be used for intermediate calculations. The second one is the type of the elements in the collection (`Angle`). The function returns the reduced value (`double`). In our example, the function passed as a parameter receives a `double` and an `Angle` parameter and returns a double.
   - Test it to compute the summation of all the degrees in an `Angle` collection and to compute its maximum sine.
   - Note: To perform the following test, an optional parameter must be added. Test it to compute group people by name; e.g., 10 people named María, 3 people named Pedro…)

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

Use the appropriate programming language features learned so far.