Challenge 2. Build a hierarchy of solvers for $f(x) = 0$

Luca Formaggia and Pasquale Africa

AA 21-22

We have seen that the file `basicZeroFun.hpp` contains a set of functions to find the zero of a scalar (univariate) function.
Having function is fine but not very flexible. So a possibility is to change the design and take a more Object-Oriented (OO) approach.
Therefore, the objective of this challenge is to build a hierarchy of solvers that roughly follows the structure indicated in the figure (it is only a suggestion!).

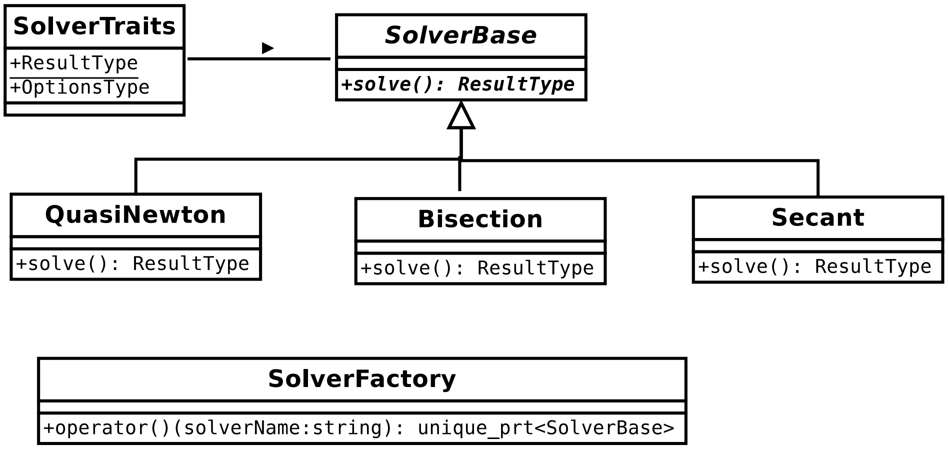

Figure 1: Possible class layout

We have an abstract class that exposes the pure virtual method solve() that takes no argument and return the results. All the other information needed for the class is stored as data member and given either via the constructor or via getters (not indicated in the sketch given in the figure.)
A *trait* collects the main types used by all classes, to allow more flexibility and readability.
A SolverFactory provides a SolverBase resource.
Write also a main() where you show the use to find the zero of $f(x) = 0.5 - e^{\pi x}$ (of which, clearly, we can get the exact solution).
Please comment the class and the methods.

1 Suggestions and available software

General rules

- The files you return include header and source files, a Makefile, possibly a README file with comments. But **NOT** executables, libraries or object files. Those should be regenerated by the Makefile.

- If you go in any directory of the examples that contains significant code and type `make install`, you copy the header files in `Examples/include` and the possible libraries in `Examples/lib`. So, if you want to use utilities in the `Example` structure, instead of copying them locally, you just have to indicate those directories in the `Makefile`, for the header and libraries, respectively.

Some suggestions

- We have shown only 3 methods, but you can add others. And also choose other names for the classes.

- Maybe you want to use bracketInterval to make sure that the interval passed to the bisection or Brent, or regula falsi algorithms does indeed bracket a zero. Beware however that in this case you need to do something to handle the situation when the function is not crossing the zero and the user by mistake has asked, for instance, to use bisection.

- You should write some comments in your code to help us understand what you are doing...

2 General rules for the Challenges

- Challenges are meant to be a tool to help you exercise programming, put in practice what seen at the lecture, and getting ready for the project. Not really to grade you. They are graded in order to give some satisfaction to the ones of you who is making the effort, but this is not their main purpose.

  So, if you need help, ask for it. Use the Forum on `WeBeep`, so that the answer may be useful also to others.

- When finished, put everything in a compressed file (use zip, or tgz, or 7z, we don't care) and upload it in `WeBeep`.