Munro Library Challenge

Provided in this archive is a CSV file containing information about munros and munro tops within Scotland. The goal of your solution is to create a library which other software can use to sort and filter the munro data. Your solution can use any programming language but we would prefer you to use a language relevant to the position you are applying for:

- Kotlin/Java for Android roles
- C# for Xamarin roles
- Swift for iOS roles

The library should provide the following functionality:

- Filtering of search by hill category (i.e. Munro, Munro Top or either). If this information is not provided by the user it should default to either. This should use the “post 1997” column and if it is blank the hill should be always excluded from results.
- The ability to sort the results by height in meters and alphabetically by name. For both options it should be possibly to specify if this should be done in ascending or descending order.
- The ability to limit the total number of results returned, e.g. only show the top 10
- The ability to specify a minimum height in meters
- The ability to specify a maximum height in meters
- Queries may include any combination of the above features and none are mandatory.
- Suitable error handling for invalid queries (e.g. when the max height is less than the minimum height)

Optionally you may choose to include the following feature if you can think of a good approach but it is not required to complete the solution. Correctness and structure of the rest of your solution is more important than adding this extra objective:

- The ability to combine sort criteria in order of preference. For example: sort by height descending and then alphabetically by name as a secondary criteria (for when the height is the same)

The query results should be returned as a list of items. Each item should contain the name, height in meters, hill category and grid reference (e.g. NN773308). Other fields should not be included.

[NOTES ON NEXT PAGE]

Notes:

- Your solution should **not** use a database and should be designed so that an amended version of the CSV file could easily be loaded.
- No extra libraries should be used for the main code of the solution but you can use any part of the standard library.
- Use of libraries is allowed in any testing/verification code. Please include any testing code you write in your submitted solution.
- Whilst developing your solution please commit your work into a git repository as you go. This is not to see how much time you take or at what times you worked on the solution but is so that we can evaluate how you broke down and approached the problem.
- There is no need to provide a GUI for the project or command line interface for this project.