

Questions

Which parts of the application would you have implemented differently and why, should you have more time?

I would implemented the cache system of the application using a database. This will remove from the application the task of store data. Since the expiration time of a Entry in the database is pretty fast (1 minute) I thought about a cron job which can remove expired prediction from database.

There is still some duplicated code between the Temperature Clients that can be moved to the BaseClient.

Not all input values are validated. For example is it possible to enter a date in the incorrect format and the application will still go on without raise exceptions. I implemented just the validation requested in the assignment. In a real life case would be crucial to prevent all kind of wrong inputs.

Performance wise, I see the application consists in multiple loops. An example is the double loop in the TemperatureReducer. I would try to implement the same using less nested loops.

If we would have asked you to unit test your code with 80% code coverage, which parts would you cover, which not and why. How would you be able to verify and prove to us that you achieve such coverage?

Just unit tests for 3 classes are present. I tried to set up 3 different kind of test in order to show I would behave with different type of classes and tests.

Code coverage can be verified and proved using any coverage tool, for example Xdebug.

I would left out from the tests the classes in the model folder, the only methods there are getters and setters.