

Design Patterns

Laboratory 2

Problem

Bank Management Propose a model for managing a bank. An empirical model for the system contains the following entities:

- a) Bank – that contains a list of clients and a bank code
- b) Bank account - that has a number and an amount of money
- c) Client – that has a name, address, a list of accounts (minimum one, maximum five)

The accounts can have RON and EUR currency. For RON accounts the interest is calculated as follow: 0.3 RON/day for amounts less than 500 RON and 0.8 RON/day for bigger amounts. The EUR accounts the interest is 0.1 EUR/day. The allowed operations on account are depositing and retrieving money.

REQUIREMENTS

Starting from the modification done to the project in previous laboratory, add the following functionality:

- Implement builder pattern for Client class in order to allow banks to store additional information like client date of birth
- Decide which pattern is most suitable for creating accounts: factory method or abstract factory. Implement the chosen pattern
- Add *close account* operation to the list of operations that can be done on an account
- Refactor the banks creation in order to use a modified singleton pattern to have only one instance for each bank present in application

OBS

At the end of the laboratory present the modification done in the project in order to obtain laboratory evaluation mark , if not all the requirements are implemented present at the next laboratory the project