

Universitatea Tehnica din Cluj-Napoca
Departament Calculatoare
Tehnici de Programare, 2018 / 2019

Prof. Ioan Salomie, Dr. Tudor Cioara, Dr. Cristina Pop
ioan.salomie@cs.utcluj.ro, tudor.cioara@cs.utcluj.ro, cristina.pop@cs.utcluj.ro

Homework 6

Objective

Review

Description

Design and implement a banking management application that allows an administrator to define bank customers, to associate multiple accounts to each customer, to add/remove accounts and to filter the accounts. There are two types of bank accounts, SavingAccount and SpendingAccount.

Requirements:

1. Draw the UML class diagram taking into account the requirements below.
2. Define the interface BankProc (add/remove persons, add/remove holder associated accounts, read/write accounts data, report generators, etc). Specify the pre and post conditions for the interface methods.
3. Define and implement the classes Person, Account, SavingAccount and SpendingAccount. Other classes may be added as needed (give reasons for the new added classes).
4. An Observer DP will be defined and implemented. It will notify the account main holder about any account related operation.
5. Implement the class Bank using a predefined collection which uses a hashtable. The hashtable key will be generated based on the account main holder (ro. titularul contului). A person may act as main holder for many accounts.
 - 4.1 Define a method of type “well formed” for the class Bank.
 - 4.2 Implement the class using Design by Contract method (involving pre, post conditions, invariants, and assertions).
6. Use stream processing to filter all accounts of type SavingAccount that have a deposit larger than 50.000, and count how many of these accounts have deposits with each multiple of 10K by collecting them in a map of type <Integer, Integer> (e.g. 5 accounts in interval 50K-60K, 7 accounts in interval 60K-70K, etc.)
7. Implement a test driver for the system.