Université Grenoble Alpes 2018-2019
M1 MSIAM

Algorithms and software tools - C++
Lab 5

`

The exercises of this Lab improve the solutions of Lab 4 using polymorphism.

Exercise 1.

We want to improve the classes that represent the ***banking accounts*** so that:

- *debiting* an amount of money from a account of the most general class cannot actually be defined (no relevant definition), this functionality is relevant only for current accounts and savings accounts
- it is possible to *display* the attributes of any kind of account by means of the stream insertion operator <<

1. Why is polymorphism useful? Which declarations and definitions will have to be modified?

2. Modify the files `account.h` and `account.cpp` accordingly, and check this new version.

Exercise 2.

Now a ***client*** will be characterized by his *name* (a string), his *identifier* (an unsigned integer), a *dynamic array of pointers to his accounts* (whatever the type of the account is), the *maximum number* of accounts, and the *current number* of accounts.
When a new client will be created, his name, identifier, and maximum number of accounts will be given.
For every client, a method *CreateAccount* will enable to create and insert in the array a new account (of any type, specified interactively by the user). Every newly created account will have a balance of 0.
It will also be possible to: credit any of his accounts number *k* (*k* is the index in the array) of an amount of money *a*, try to debit any of his accounts number *k* of an amount of money *a*, and display his attributes (including the attributes of his accounts).

Here is a example of expected interaction, for the creation of accounts for Smith, and displaying the attributes of Smith:

```
Creation of accounts for Smith:
Creation of a current account(1), unblocked savings account(2) or blocked
savings account(3) ?
2
Creation of a current account(1), unblocked savings account(2) or blocked
savings account(3) ?
1

Client name : Smith and id = 1234 and his/her accounts :
** Savings account **
Account number = 0 , balance = 0 euros
and Interest rate = 2%
** Current account **
Account number = 1 , balance = 0 euros
```

1. Give the declaration of this new class Client: modify/add attributes and methods.

2. Define all the methods and functions declared in question 1.

3. Define a copy constructor for the class Client.

4. The class Bank is the same as in Lab 4. Adapt your *main* function to use all your new classes.