Bank Project

Used functionalities

- Interface
- Enumeration
- Exception, try/catch
- Abstract types (E)

Description

- We manage a bank where customers have accounts.
- They can create an account, add money on it and get money out.
- They can as well transfer money to another account
- All the operations can be performed either in the bank or online

Implementation

Create an interface Bank which has the following methods:
a. void transfer (double amount, Account fromAccount, Account toAccount)
b. void withdraw (double amount, Account fromAccount)
c. void deposit(double amount, Account toAccount)
d. bool checkIdentity(E identity)
e. Account createAccount(double initialAmount, Customer customer, String accountName)

Implements the interface in the objects Bancomat and EBanking

- In the real bank, the identity is tested with a pin code (int) and online with a password (String)
- You cannot transfer money or create an account from a bancomat

For each operation

- Check the identity of the customer, raise an exception if failed (for account creation as well).
  The pin or password should be a user input from console
  When a customer is locked, he can only receive money
- No negative sold on any account is allowed, this should raise an exception as well
- In case of error the balance of the account should be the same as before the transaction

General rules

- The customer should never see an Exception, instead they should be try/catch, and the user must be informed on what happened
- Three error in the identification should lock the customer.
  The different state should be managed with an enumeration (locked, ok).
- The customer should always receive feedbacks after each operation, including identification
- You should not be able to transfer money from/to the same account
- Account name should be unique by customer