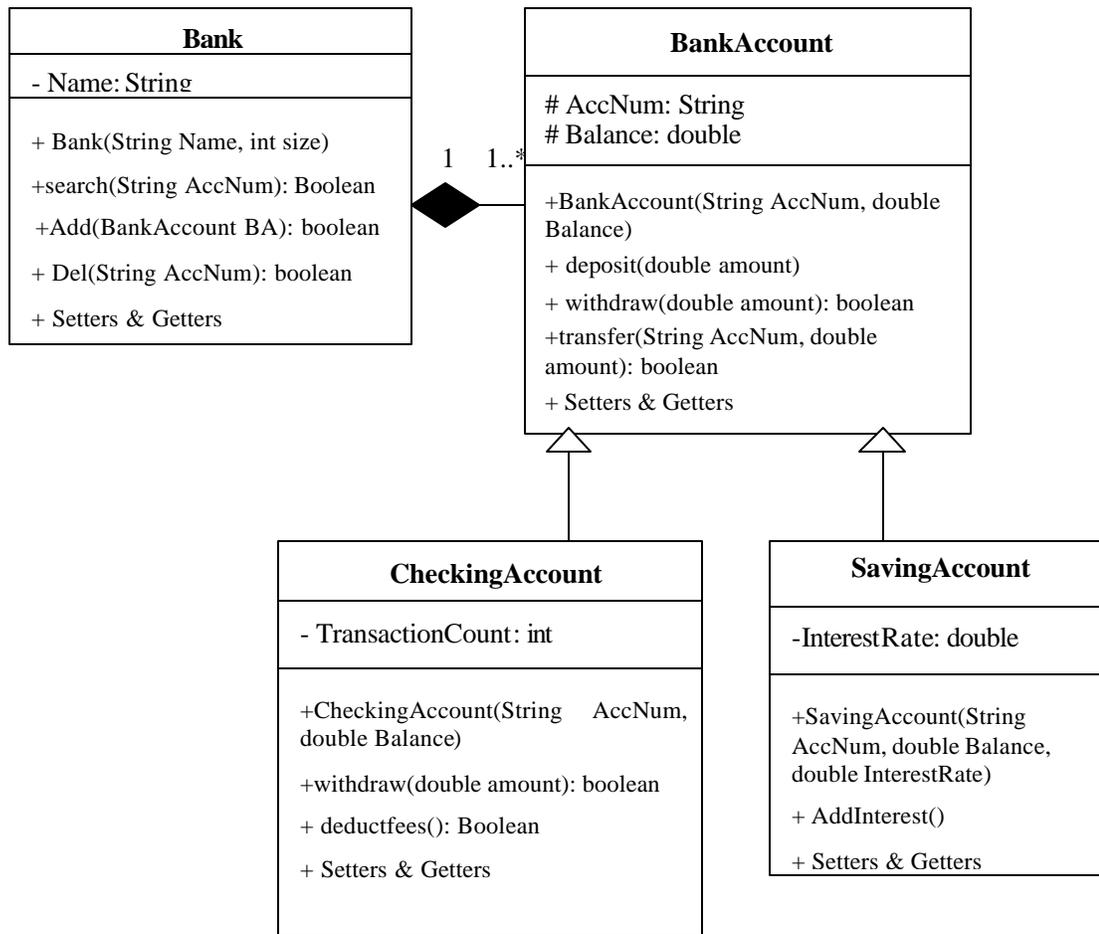


Lab No. – Exception Handling

Date: 08 – 12/ 12 / 2007

A bank manages many customers' accounts. Each bank account can be a saving or a checking account. The UML diagram of this application is presented in the following figure.



BankAccount Class.

The class BankAccount provides three services. A function deposit to add an amount to the current balance. A function withdraw to subtract money from the account and ensure that the debit amount does not exceed the account's balance. A transfer function that deposit money to another account.

CheckingAccount Class.

The class CheckingAccount should inherit from base class BankAccount and include an additional data member TransactionCount that represents the number of transaction processed. This class should redefine function withdraw and implements a new function deductfees that withdraw transactions fees from the account balance whenever the transaction is performed successfully and the number of transaction exceed 3. Such that the fee of one transaction is equal 2 riyals.

SavingAccount Class.

The savingAccount class inherits from the BankAccount class and includes a new attribute InterestRate indicating the interest rate assigned to the account. The function AddInterest should deposit the amount of interest earned by an account.

Bank Class.

The Bank class implements the following functions:

- The constructor Bank(String Name, int size) and throws an exception when the array is created with a negative size. (**NegativeArraySizeException**)
- The function Search to search an account and returns -1 if is not founded and his index otherwise.
- The function Add to add a new account and throws an exception when an illegal index term in the array has been accessed. (**ArrayIndexOutOfBoundsException**)
- The function Del to remove an account and throws an exception when the method has been passed an illegal or inappropriate argument. (**IllegalArgumentException**)

Question

Implement the UML diagram presented previously and write a program that processes a Bank object according to the following menu. Such that when user tapes 1, the program should display another sub menu concerning only account operations. Throw an exception when the user enter a choice different of the type Integer. (**InputMismatchException**)

