CS1301 Programming Assignment#4 Due on Friday, 9/4/2015 at 10:00PM

9/1/2015 MyAccount4 UML.

Fares

1. Implement MyAccount4 class as specified in the UML diagram below.
2. Make sure you use the methods, arguments, and the return types exactly as specified in the UML.
3. Create a driver (a class with a main method) to create objects of MyAccount4 class and invokes its methods. You may name it MyAccount4Tester.
4. Make sure you document your class, methods, and code.
5. Attach (do not turn in any hardcopy) via BlazeView the following items:
   1. A single typed page with your name, class, date, and program title. The report (named MyReport4) should include what you learned from the programming assignment, problems faced, skills learned, and your observations. Also, append to your report the following fill in self-evaluation:
      1. Have you documented your MyAccount4 class: \_\_\_\_\_?
      2. Have you documented its methods: \_\_\_\_\_\_\_?
      3. Is your program well structured, aligned, indented, and easy to read: \_\_\_\_\_?
      4. Does your class MyAccount4 compile without syntax errors: \_\_\_\_\_\_\_\_\_\_\_?
      5. Does your driver program run without runtime errors: \_\_\_\_\_\_\_\_\_\_\_?
      6. Does your program run without logical errors: \_\_\_\_\_\_\_\_\_\_\_?
   2. Even though I will use my own driver to test your MyAccount4, You need to submit you MyAccount4, MyAccount4Tester, and report.
   3. Zip all your files into single one. Name it with your last name followed by the program#. For example, Fares4.
6. Implement the UML diagram below.

|  |  |
| --- | --- |
| MyAccount4 | Documentation |
| //Instance Variables (name followed by type)  -name: String  - accountNumber:long  -balance:double  -dateCreated:Date | Owner’s name  Account Number  Account Balance  Date of account created |
| //Constructors & methods (name followed by type)  +MyAccount4()  +MyAccount4(owner:String, accountNumber:long,  balance:double)  //Getters & Setters  +getName():String  +getAccountNumber():long  +getBalance():double  +setName(name:String):void  +deposit(amount:double):void  +withdraw(amount:double):void  +toString():String | The no-argument constructor assigns “Unknown” to name, 99999 to accountNumber, 0 to balance, and get the date from the system  The overloaded constructor assigns owner to name, account Number to accountNumber, balance to balance, and get the date from the system  Returns the owner’s name  Returns the account number  Returns the balance  Updates the name  Adds deposit amount to balance  Subtracts withdrawal from balance  Returns name, account number, balance, and date created. Make sure you include labels and print them on separate lines with appropriate indentions and alignment. |