Mariam Tsirekidze

ID:823460489

Prof: M. Tsintsadze

Report

Assignment #3 “Bank Account”

Problem:

Bank Financial Systems, Inc., develops financial software for banks and credit unions. The company is developing a new system that manages customer accounts. One of your tasks is to develop a class that represents a savings account. The data that must be held by an object of this class is

- The account number
- The interest rate
- The account balance

You must also develop a class that represents a certificate of deposit (CD) account. The data that must be held by an object of this class is

- The account number
- The interest rate
- The account balance
- The account maturity date

Create a SavingsAccount class to represent a savings account and then create a class that is derived from SavingsAccount, named CDAccount, to represent a CD account.

Implement the methods deposit() and withdraw() and the method toString() to obtain a string containing the information on a CD account.

To test the classes, let the user enter data about a CD account, create an object of the CDAccount class, set the object’s properties to the data that the user entered, and then display the object’s data. Give possibility to user to deposit and withdraw amounts correctly (keep in mind the balance)

Create the list of accounts where all accounts will be stored and display them in a descending order based on their balance

Solution:

Class SavingsAccount:

Three data variables for:

- " **string AccountNum**" – Account number
- "**double rate**" – rate
- "**double balance**" – balance

public SavingsAccount(string accountNum, double rate, double balance) – constructor

public SavingsAccount() –default constructor

public void deposite(double depos) – function when the customer wants to add money on the account

public void withdrow(double withdr) – function when the customer wants to withdraw money from the account. If there is not enough money on the account the customer will receive message and function won't decrease balance.

class CDAccount : SavingsAccount: is derived class from the SavingsAccount

one private variable for Date

- "**DateTime date**" – account maturity date

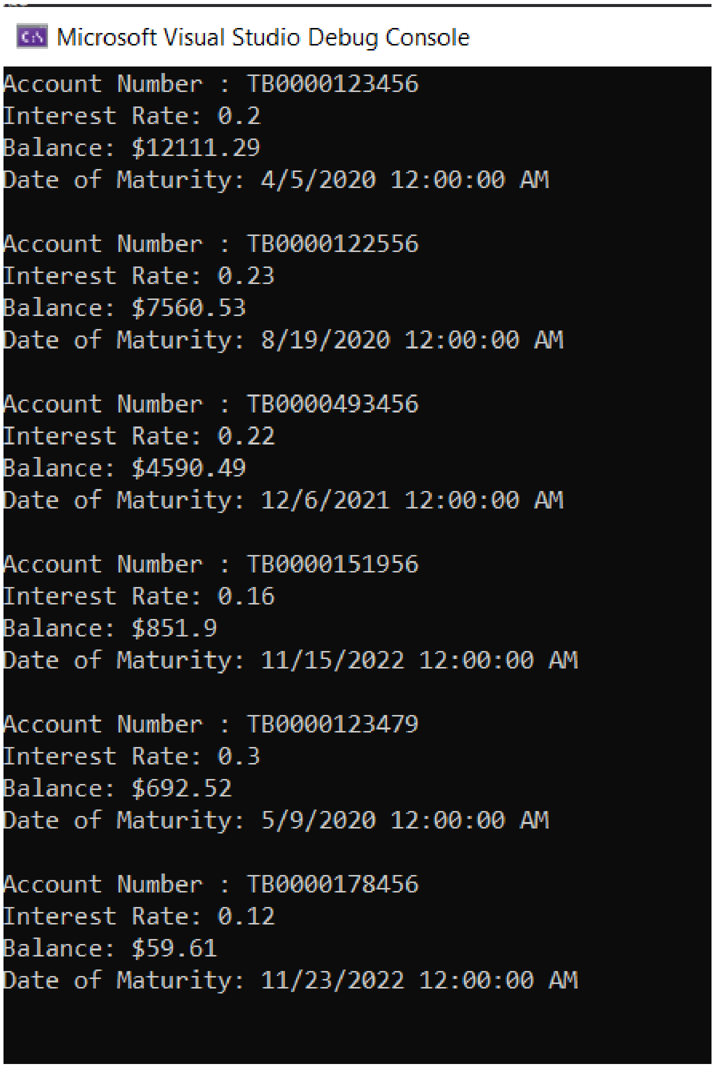
public CDAccount(string AccountNum, double Rate, double Balance, DateTime Date)

: base(AccountNum, Rate, Balance) – constructor with call base class SavingsAccount constructor.

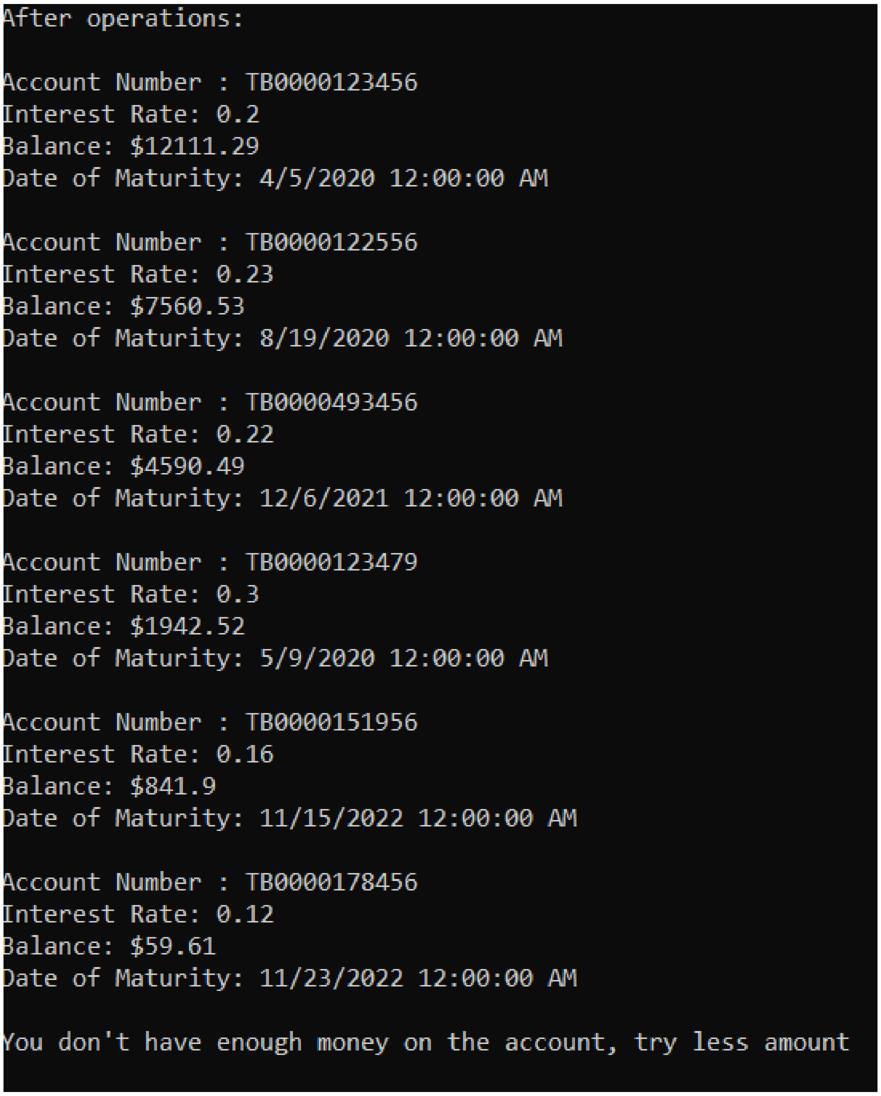
public CDAccount() : base() – default constructor

public override string ToString() – for string representation of CDAccount objects

in Main function I am creating lists of the CDAccount objects and add multiple objects with it's different values in it. Then I am sorting the list by descending balance values with the help of LINQ. Then display all the list objects. Then I am trying to withdraw and deposit money on accounts and then again display the list of objects, then I am trying to take money that is more then balance to see the warning message.



```
Account Number : TB0000123456
Interest Rate: 0.2
Balance: $12111.29
Date of Maturity: 4/5/2020 12:00:00 AM

Account Number : TB0000122556
Interest Rate: 0.23
Balance: $7560.53
Date of Maturity: 8/19/2020 12:00:00 AM

Account Number : TB0000493456
Interest Rate: 0.22
Balance: $4590.49
Date of Maturity: 12/6/2021 12:00:00 AM

Account Number : TB0000151956
Interest Rate: 0.16
Balance: $851.9
Date of Maturity: 11/15/2022 12:00:00 AM

Account Number : TB0000123479
Interest Rate: 0.3
Balance: $692.52
Date of Maturity: 5/9/2020 12:00:00 AM

Account Number : TB0000178456
Interest Rate: 0.12
Balance: $59.61
Date of Maturity: 11/23/2022 12:00:00 AM
```



```
After operations:

Account Number : TB0000123456
Interest Rate: 0.2
Balance: $12111.29
Date of Maturity: 4/5/2020 12:00:00 AM

Account Number : TB0000122556
Interest Rate: 0.23
Balance: $7560.53
Date of Maturity: 8/19/2020 12:00:00 AM

Account Number : TB0000493456
Interest Rate: 0.22
Balance: $4590.49
Date of Maturity: 12/6/2021 12:00:00 AM

Account Number : TB0000123479
Interest Rate: 0.3
Balance: $1942.52
Date of Maturity: 5/9/2020 12:00:00 AM

Account Number : TB0000151956
Interest Rate: 0.16
Balance: $841.9
Date of Maturity: 11/15/2022 12:00:00 AM

Account Number : TB0000178456
Interest Rate: 0.12
Balance: $59.61
Date of Maturity: 11/23/2022 12:00:00 AM

You don't have enough money on the account, try less amount
```

```
//Tsirekidze Mariam
//ID:823460489

using System;
using System.Collections.Generic;
using System.Text;

namespace BankAccount
{
    class CDAccount : SavingsAccount
    {
        private DateTime date;

        //constructor with call base class SavingsAccount constructor
        public CDAccount(string AccountNum, double Rate, double Balance, DateTime Date)
            : base(AccountNum, Rate, Balance)
        {
            date = Date;
        }

        //default constructor
        public CDAccount() : base()
        {
            date = new DateTime();
        }
        //return string representation of CDAccount object
        public override string ToString() =>
            $"Account Number : {AccountNum} \n" +
            $"Interest Rate: {Rate}\n" +
            $"Balance: ${Balance}\n" +
            $"Date of Maturity: {date}\n";




    }
}
```

```
//Tsirekidze Mariam
//ID:823460489
using System;
using System.Collections.Generic;
using System.Text;

namespace BankAccount
{
    class SavingsAccount
    {
        public string AccountNum { get; set; }  // properties
        public double Rate { get; set; }
        public double Balance { get; set; }

        public SavingsAccount(string accountNum, double rate, double balance) //constructor
        {
            AccountNum = accountNum;
            Rate = rate;
            Balance = balance;
        }

        public SavingsAccount() //default constructor
        {
            AccountNum = null;
            Rate = 0.0;
            Balance = 0.0;
        }
        public void deposite(double depos) //increase balance function
        {
            Balance = Balance + depos;
        }
        public void withdrow(double withdr) //decrease balance function
        {
            if(withdr>Balance)
            {
                Console.WriteLine("You don't have enough money on the account, try less amount\n");
            }
            else
            {
                Balance = Balance - withdr;
            }

        }
    }
}
```

```
//Tsirekidze Mariam
//ID:823460489
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices.ComTypes;

namespace BankAccount
{
    class Program
    {
        static void Main(string[] args)
        {

            //create list of CDAccounts
            List<CDAccount> listOfAcc = new List<CDAccount>();

            //add accounts
            listOfAcc.Add(new CDAccount("TB0000123456", 0.2, 12111.29, new DateTime(2020, 04, 5)));
            listOfAcc.Add(new CDAccount("TB0000123479", 0.3, 692.52, new DateTime(2020, 05, 9)));
            listOfAcc.Add(new CDAccount("TB0000178456", 0.12, 59.61, new DateTime(2022, 11, 23)));
            listOfAcc.Add(new CDAccount("TB0000122556", 0.23, 7560.53, new DateTime(2020, 08, 19)));
            listOfAcc.Add(new CDAccount("TB0000493456", 0.22, 4590.49, new DateTime(2021, 12, 6)));
            listOfAcc.Add(new CDAccount("TB0000151956", 0.16, 851.9, new DateTime(2022, 11, 15)));



            //sort list of accounts by desceding balance
```

```
var filtered =
    from element in listOfAcc
    orderby element.Balance descending
    select element;

//output the accounts
foreach(var item in filtered )
{
    Console.WriteLine(item);
}

//check the depostide and withdraw functions if they work
listOfAcc[1].deposite(1250);
listOfAcc[5].withdrow(10);

//sort list of accounts by desceding balance
var filtered2 =
    from element in listOfAcc
    orderby element.Balance descending
    select element;



Console.WriteLine("\nAfter operations:\n");
//output the accounts
foreach (var item in filtered2)
{
    Console.WriteLine(item);
}
```

```
            listOfAcc[5].withdrow(10000);




        }

    }

}
```