Deakin University

Object Oriented Development

OnTrack Submission

The Account Class

Submitted By:
Peter Stacey
pstacey
2020/03/19 07:17

Tutor:
Dipto Pratyaksa

Outcome	**Weight**
Evaluate Code	♦◇◇◇◇
Principles	♦◇◇◇◇
Build Programs	♦♦◇◇◇
Design	♦♦◇◇◇
Justify	♦◇◇◇◇

At this point, being a task that we will continue to build on throughout the semester, the program didn't require much design, although it currently has some significant limitations (eg. no protection against overdrawing the account) that need to be fixed in the future. It was more, the beginning of a larger program and straight forward to implement.

March 19, 2020

```
using System;

namespace Task_2._2P
{
    class TestAccount
    {
        static void Main(string[] args)
        {
            Console.WriteLine("\n**************");
            Console.WriteLine("TESTING START");
            Console.WriteLine("**************");

            Console.WriteLine("\n----------------------");
            Console.WriteLine("GOOD ACCOUNT BEHAVIOUR");
            Console.WriteLine("----------------------");

            Account okAccount = new Account("Mrs Good", 0);

            okAccount.Print(); // Expect balance to be $0.00
            okAccount.Deposit(500);
            okAccount.Withdraw(100);
            okAccount.Print(); // Expect balance to be $400.00
            Console.WriteLine("Account name: " + okAccount.Name);

            Console.WriteLine("\n----------------------");
            Console.WriteLine("BAD ACCOUNT BEHAVIOUR");
            Console.WriteLine("----------------------");

            Account badAccount = new Account("Mr Bad", -100); // Allows a negative
            ↪  balance

            badAccount.Print(); // Expect balance to be -$100.00
            badAccount.Deposit(100);
            badAccount.Print(); // Expect balance to be $0.00
            badAccount.Withdraw(1000000000); // Expect $1 billion overdrawn
            badAccount.Print();
            // badAccount.Name = "I'm really ok"; // Confirm read-only

            Console.WriteLine("\n--------------------");
            Console.WriteLine("ATTEMPTED BEHAVIOUR");
            Console.WriteLine("--------------------");

            Account terribleAccount = new Account("okAccount.Withdraw(1000);", 0);
            ↪  // Attempting to affect ok account
            terribleAccount.Print();
            okAccount.Print(); // Expect $400.00

            Console.WriteLine("\n************");
            Console.WriteLine("TESTING END");
            Console.WriteLine("************\n");

            Console.ReadLine();
        }
```

```
52      }
53  }
```

```cs
using System;

namespace Task_2._2P
{
    /// <summary>
    /// A bank account class to hold the account name and balance details
    /// </summary>
    class Account
    {
        // Instance variables
        private String _name;
        private decimal _balance;

        // Read-only properties
        public String Name { get { return _name; } }


        /// <summary>
        /// Class constructor
        /// </summary>
        /// <param name="name">The name string for the account</param>
        /// <param name="balance">The decimal balance of the account</param>
        public Account(String name, decimal balance)
        {
            _name = name;
            _balance = balance;  // !Allows negative initial balance
        }

        /// <summary>
        /// Deposits money into the account
        /// </summary>
        /// <param name="amount">The decimal amount to add to the balance</param>
        public void Deposit(decimal amount)
        {
            _balance += amount;
        }

        /// <summary>
        /// Withdraws money from the account (with no overdraw protection currently)
        /// </summary>
        /// <param name="amount">The amount to subtract from the balance</param>
        public void Withdraw(decimal amount)
        {
            _balance -= amount; // !Allows unlimited overdraw
        }

        /// <summary>
        /// Outputs the account name and current balance as a string
        /// </summary>
        public void Print()
        {
            Console.WriteLine("Account Name: {0}, Balance: {1}",
                _name, _balance.ToString("C"));
```

```
54            }
55        }
56    }
```