Mansoura University

Faculty of Computers and Information

Department of Computer Science

First Semester - 2021 -2022

Object Oriented Programming Project

Supervisor:

Prof. Osama Abuelnasr

Team members:

Mohamed Saad Ahmed Ali Omar, SWE, 805554525

Farah Aysam Mohamed Elsaid, SWE, 805481657

Mariam Abdelnaby hessein Elsaid, SWE, 805775545

Project Idea:

A bank with a checking, or 25% interest rate saving account

Run Example

```
❯ dotnet run
Choose a bank type 1 or 2
1- Checking account
2- Saving account
1
Enter your inital deposit: 1000
--------------------
What's your next move
1- Deposit
2- Withdraw
3- Check balance
4- Switch account type
5- Exit
Move number: 1
Enter your deposit amount: 1000
The account balance is: 2000
In text: two Thousand
--------------------
What's your next move
1- Deposit
2- Withdraw
3- Check balance
4- Switch account type
5- Exit
Move number: 4
The account balance is: 2500
In text: two Thousand and more
--------------------
--------------------
What's your next move
1- Deposit
2- Withdraw
3- Check balance
4- Switch account type
5- Exit
Move number: 5
bye
```

```
// Mohamed Saad Ahmed Ali Omar, SWE, 805554525
// Farah Aysam Mohamed Ẹlsaid, SWE, 805481657
// Mariam Abdelnaby hessein Elsaid, SWE, 805775545
using System;

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

            // Ask the user for the bank account type
            Console.WriteLine("Choose a bank type 1 or 2");
            Console.WriteLine("1- Checking account");
            Console.WriteLine("2- Saving account");

            int type = Convert.ToInt32(Console.ReadLine());

            if (type == 1)
            {
                Console.Write("Enter your inital deposit: ");
                int number = Convert.ToInt32(Console.ReadLine());
                Account CAccount = new CheckingAccount(number);
                login.log(CAccount);
            }

            else if (type == 2)
            {
                Console.Write("Enter your inital deposit: ");
                int number = Convert.ToInt32(Console.ReadLine());
                Account SAccount = new SavingsAccount(number);
                login.log(SAccount);
            }
            else
            {
                Console.WriteLine("Wrong Input");
            }

        }

    }

}
```

```cs
using System;
namespace OOP_Project_Bank
{
    interface IAccount
    {
        bool Deposit(double amount);
        bool Withdraw(double amount);
        void PrintBalance();

    }
    class Account : IAccount
    {
        public string accountType;

        private double _balance;

        protected double Balance
        {
            get { return this._balance; }
            set
            {
                if (value >= 0)
                    this._balance = value;
            }
        }

        // Constructor
        public Account(double balance)
        {
            this.Balance = balance;
        }

        public Account(Account switchedFrom)
        {
            this.Balance = switchedFrom.Balance;
        }

        //Virtual Methods
        public virtual bool Deposit(double amount)
        {

            return false;
        }

        public virtual bool Withdraw(double amount)
        {

            return false;
        }

        public virtual void PrintBalance()
        {
            // Console.WriteLine("The balance is: " + Balance);
            Console.WriteLine("In text: " + Balance.ToString());
        }

    }
}


```

```
using System;
namespace OOP_Project_Bank
{
    class SavingsAccount : Account
    {
        private double _interestRate;

        // Copy Constructor
        public SavingsAccount(Account switchedFrom) : base(switchedFrom)
        {
            accountType = "Saving";
            this._interestRate = 0.25;
            Balance += (Balance * this._interestRate);
        }

        // Paramterized Constructor
        public SavingsAccount(double balance)
            : base(balance)
        {
            this._interestRate = 0.25;
            Balance += (Balance * this._interestRate);
        }

        public override bool Deposit(double amount)
        {
            if (amount > 0)
            {
                Balance += amount + (amount * this._interestRate);
                return true;
            }
            return false;
        }


        public override bool Withdraw(double amount)
        {
            if (amount > 0 && amount <= Balance)
            {
                // Deducting from balance
                Balance -= amount;
                return true;
            }
            return false;
        }


        public override void PrintBalance()
        {

            Console.WriteLine("The checking account balance is: " + base.Balance);
            Console.WriteLine("In text: " + ToString());
        }

        public override string ToString()
        {
            string[] numbers = { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" };
            string[] tens = { "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen" };
            string[] ten = { "zero", "Ten", "Twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety" };
            string[] levels = { "Hundred", "Thousand", "Million" };
            string strYear = Balance.ToString();

            if (strYear.Length == 0)
            {
                return numbers[0];
            }
            else if (strYear.Length == 1)
            {
                int num = Convert.ToInt32(strYear[0]) - '0';
                return numbers[num];
            }
            else if (strYear.Length == 2)
            {
                if (Balance < 20 && Balance > 9)
                {

                    int num = Convert.ToInt32(strYear[1]) - '0';
                    return tens[num];
                }
                else
                {
                    int tenat = Convert.ToInt32(strYear[0]) - '0';
                    int num = Convert.ToInt32(strYear[1]) - '0';
                    return ten[tenat] + " " + numbers[num];
                }
            }
            else if (strYear.Length == 3)
            {
                int num = Convert.ToInt32(strYear[0]) - '0';
                int tenat = Convert.ToInt32(strYear[1]) - '0';
                int ones = Convert.ToInt32(strYear[2]) - '0';
                return numbers[num] + ten[tenat] + " " + numbers[num];
            }
            else if (strYear.Length == 4)
            {
                int thau = Convert.ToInt32(strYear[0]) - '0';
                int hun = Convert.ToInt32(strYear[1]) - '0';
                int tenat = Convert.ToInt32(strYear[2]) - '0';
                int ones = Convert.ToInt32(strYear[3]) - '0';

                if (hun > 0 || tenat > 0 || ones > 0)
                    return numbers[thau] + " " + levels[1] + " and more";
                else
                    return numbers[thau] + " " + levels[1];
            }
            else
            {
                return "more than 9999";
            }
        }
    }
}
```

```
using System;
namespace OOP_Project_Bank
{
    class CheckingAccount : Account
    {

        // Copy Constructor
        public CheckingAccount(Account switchedFrom) : base(switchedFrom)
        {
        }

        // Paramterized Constructor
        public CheckingAccount(double balance)
            : base(balance) { accountType = "Checking"; }


        public override bool Deposit(double amount)
        {
            if (amount > 0)
            {
                Balance += amount;
                return true;
            }
            return false;
        }


        public override bool Withdraw(double amount)
        {
            if (amount > 0 && amount <= Balance)
            {
                Balance -= amount;
                return true;
            }
            return false;
        }


        public override void PrintBalance()
        {

            Console.WriteLine("The checking account balance is: " + base.Balance);
            Console.WriteLine("In text: " + ToString());
        }

        public override string ToString()
        {
            string[] numbers = { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" };
            string[] tens = { "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen" };
            string[] ten = { "zero", "Ten", "Twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety" };
            string[] levels = { "Hundred", "Thousand", "Million" };
            string strYear = Convert.ToString(Balance);

            if (strYear.Length == 0)
            {
                return numbers[0];
            }
            else if (strYear.Length == 1)
            {
                int num = Convert.ToInt32(strYear[0]) - '0';
                return numbers[num];
            }
            else if (strYear.Length == 2)
            {
                if (Balance < 20 && Balance > 9)
                {

                    int num = Convert.ToInt32(strYear[1]) - '0';
                    return tens[num];
                }
                else
                {
                    int tenat = Convert.ToInt32(strYear[0]) - '0';
                    int num = Convert.ToInt32(strYear[1]) - '0';
                    return ten[tenat] + " " + numbers[num];
                }
            }
            else if (strYear.Length == 3)
            {
                int num = Convert.ToInt32(strYear[0]) - '0';
                int tenat = Convert.ToInt32(strYear[1]) - '0';
                int ones = Convert.ToInt32(strYear[2]) - '0';
                return numbers[num] + ten[tenat] + " " + numbers[num];
            }
            else if (strYear.Length == 4)
            {
                int thau = Convert.ToInt32(strYear[0]) - '0';
                int hun = Convert.ToInt32(strYear[1]) - '0';
                int tenat = Convert.ToInt32(strYear[2]) - '0';
                int ones = Convert.ToInt32(strYear[3]) - '0';

                if (hun > 0 || tenat > 0 || ones > 0)
                    return numbers[thau] + " " + levels[1] + " and more";
                else
                    return numbers[thau] + " " + levels[1];
            }
            else
            {
                return "more than 9999";
            }
        }


    }
}
```

```cs
using System;
namespace OOP_Project_Bank
{
    class login
    {
        public static void log(Account Account)
        {
            Console.WriteLine("--------------------");
            Console.WriteLine("What's your next move");
            Console.WriteLine("1- Deposit");
            Console.WriteLine("2- Withdraw");
            Console.WriteLine("3- Check balance");
            Console.WriteLine("4- Switch account type");
            Console.WriteLine("5- Exit");

            Console.Write("Move number: ");


            int opr = Convert.ToInt32(Console.ReadLine());
            if (opr == 1)
            {
                Console.Write("Enter your deposit amount: ");
                int amount = Convert.ToInt32(Console.ReadLine());
                Account.Deposit(amount);
                Account.PrintBalance();
                log(Account);
            }
            else if (opr == 2)
            {
                Console.Write("Enter your withdrow amount: ");
                int amount = Convert.ToInt32(Console.ReadLine());
                Account.Withdraw(amount);
                Account.PrintBalance();
                log(Account);
            }
            else if (opr == 3)
            {
                Account.PrintBalance();
                log(Account);

            }
            else if (opr == 4)
            {
                if (Account.accountType == "Checking")
                {
                    Account SAccount = new SavingsAccount(Account);
                    SAccount.PrintBalance();
                    Console.WriteLine("--------------------");
                    login.log(SAccount);
                }
                else
                {
                    Account CAccount = new CheckingAccount(Account);
                    CAccount.PrintBalance();
                    Console.WriteLine("--------------------");
                    login.log(CAccount);

                }
            }
            else if (opr == 5)
            {
                Console.WriteLine("bye");
            }
            else
            {
                Console.WriteLine("try again");
            }
        }
    }
}


```