// Mohammed Mohsen Mohammed Ahmady

// 323232391

// Section 10

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_5\_OOP\_Section\_Assignment\_\_BankAccount\_

{

internal class Program

{

static void Main(string[] args)

{

DateTime opendate = DateTime.Now;

BankAccount b1 = new BankAccount(54865623, 1500, "mohammed", opendate);

b1.Withdraw(500);

b1.Deposite(1000);

b1.Deposite(20000000);

b1.Display();

SpecialAccount s1 = new SpecialAccount(456465454,258000,"ahmed",opendate,20000);

s1.Withdraw(2000);

s1.Display();

}

}

}

internal class BankAccount

{

private int accountNum;

private double balance;

private string customerName;

public DateTime Opendate { get; set; }

public int AccountNum

{

get { return accountNum; }

set { accountNum = value; }

}

public double Balance

{

get { return balance; }

set { balance = value; }

}

public string CustomerName

{

get { return customerName; }

set { customerName = value; }

}

public BankAccount(int accountNum, double balance, string customerName, DateTime opendate)

{

this.AccountNum = accountNum;

this.Balance = balance;

this.CustomerName = customerName;

this.Opendate = opendate;

}

public virtual void Display()

{

Console.WriteLine($"\nAccount Number: {AccountNum}\nBalance: {Balance}\nCustomer Name: {CustomerName}\nTime: {Opendate}");

Console.WriteLine();

}

public virtual void Withdraw(double amount)

{

if (balance == 0)

{

Console.WriteLine("There is no Money To Withdraw !");

return;

}

else

{

balance -= amount;

Console.WriteLine($"-- There is {amount}$ Withdrawed From Your Account At {Opendate}\n## Your Current Balance is {balance}");

}

}

public void Deposite(double amount)

{

balance += amount;

Console.WriteLine($"++ There is {amount}$ Deposited To Your Account At {Opendate}\n## Your Current Balance is {balance}");

}

}

internal class SpecialAccount : BankAccount

{

private double overlimit;

public double Overlimit

{

get { return overlimit; }

set { overlimit = value; }

}

public SpecialAccount(int accountNum, double balance, string customerName, DateTime opendate, double overlimit) : base (accountNum, balance, customerName, opendate )

{

this.Overlimit = overlimit;

Console.WriteLine("$$$ Hello And Welcome To Your \*#Special#\* Account");

}

public override void Display()

{

Console.WriteLine();

Console.WriteLine($"Account Number: {AccountNum}\nBalance: {Balance}\nCustomer Name: {CustomerName}\nTime: {Opendate}\nThe Overlimit: {Overlimit}");

Console.WriteLine();

}

public override void Withdraw(double amount)

{

if (Balance == 0)

{

Console.WriteLine("There is no Money To Withdraw !");

return;

}

else

{

if (amount < Overlimit)

{

Balance -= amount;

Console.WriteLine($"-- There is {amount}$ Withdrawed From Your \*#Special#\* Account At {Opendate}\n## Your Current Balance is {Balance}");

}

else

{

Console.WriteLine($"You Have a Withdraw Limit = {overlimit}, You Can't Excced It\n--> Try Withdrawing Fewer Amount");

}

}

}

}