

// main program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HW09\_01

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Instantiate checking object...");

Checking c1 = new Checking("Jimmy", "Broke", "Citibank", 322271724, 9833482, 400, 0.001);

Console.WriteLine(c1);

Console.WriteLine("Jimmy deposited $10.04, then withdrew $5.89");

c1.Deposit(10.04);

c1.Withdraw(5.89);

c1.CalculateInterest();

Console.WriteLine("Jimmy wrote a check for $500 to buy a new TV");

c1.WriteCheck(500);

Console.WriteLine("\nInstantiate savings object...");

Saving s1 = new Saving("Richie", "Rich", "Chase", 322271627, 1122345, 9999999.99, 0.01);

Console.WriteLine(s1);

Console.WriteLine("Richie deposited $9999, then withdrew $88888.88");

s1.Deposit(9999);

s1.Withdraw(88888.88);

s1.CalculateInterest();

Console.WriteLine("\nRichie was sympathetic to Jimmy's plight and decided to help out");

s1.TransferToChecking(9876.88, 322271724, c1);

Console.WriteLine("\nJimmy was grateful to Richie and used the money to buy a $1500 4K TV");

c1.WriteCheck(1500);

Console.WriteLine("\nJimmy returned $1000 back to Richie");

c1.TransferToSaving(1000, 322271627, s1);

Console.WriteLine("\n" + c1);

Console.WriteLine(s1);

Console.WriteLine("Press any key to continue...");

Console.ReadKey();

}

}

}

// BankAccount class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HW09\_01

{

class BankAccount

{

// data members

protected int accountNumber, routingNumber;

protected double balance, interestRate;

protected string firstName, lastName, bankName;

// constructors

public BankAccount()

{

bankName = "unknown";

routingNumber = 0;

interestRate = 0;

accountNumber = 0;

balance = 0;

firstName = "John";

lastName = "Smith";

}

public BankAccount(string first, string last, string bank, int routing, int acctNum, double bal, double apr)

{

bankName = bank;

routingNumber = routing;

accountNumber = acctNum;

firstName = first;

lastName = last;

balance = bal;

interestRate = apr;

}

public void Deposit(double deposit)

{

balance += deposit;

Console.WriteLine("Balance = {0:c2} after deposit of {1:c2}", balance, deposit);

}

public void Withdraw(double withdraw)

{

if (withdraw > balance)

Console.WriteLine("Insufficient balance for withdrawal amount of {0:c2}. Please enter small amount", withdraw);

else

{

balance -= withdraw;

Console.WriteLine("Balance = {0:c2} after withdraw of {1:c2}", balance, withdraw);

}

}

public void CalculateInterest()

{

Console.WriteLine("Yearly interest at {0}% APR = {1:c2}", interestRate\*100, interestRate\*balance);

}

public string BankName

{

get { return bankName; }

set { bankName = value; }

}

public string FirstName

{

get { return firstName; }

set { firstName = value; }

}

public string LastName

{

get { return lastName; }

set { lastName = value; }

}

public double Balance

{

get { return balance; }

set { if (value >= 0) balance = value; else balance = 0; }

}

public double InterestRate

{

get { return interestRate; }

set { if (value >= 0) interestRate = value; else interestRate = 0; }

}

public int AccountNumber

{

get { return accountNumber; }

set { if (value >= 0) accountNumber = value; else accountNumber = 0; }

}

public int RoutingNumber

{

get { return routingNumber; }

set { if (value >= 0) routingNumber = value; else routingNumber = 0; }

}

public override string ToString()

{

return string.Format("First Name: {0}\n", firstName) +

string.Format("Last Name: {0}\n", lastName) +

string.Format("Bank Name: {0}\n", bankName) +

string.Format("Routing Number: {0}\n", routingNumber) +

string.Format("Account Number: {0}\n", accountNumber) +

string.Format("Balance: {0:c2}\n", balance) +

string.Format("Interest rate: {0}%\n", interestRate\*100);

}

~BankAccount() { }

}

}

// Saving class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HW09\_01

{

class Saving: BankAccount

{

// constructors

public Saving(): base()

{ }

public Saving(string first, string last, string bank, int routing, int acctNum, double bal, double apr) : base(first, last, bank, routing, acctNum, bal, apr)

{ }

public void TransferToChecking(double amount, int routing, BankAccount Checking)

{

if (routing == Checking.RoutingNumber)

{

Console.WriteLine("{0} {1} of account number {2}", firstName, lastName, accountNumber);

if (amount <= balance)

{

balance -= amount;

Console.WriteLine("Transferred {0:c2} to checking account number {1} with routing number {2}", amount, Checking.AccountNumber, Checking.RoutingNumber);

Console.WriteLine("Old savings balance: {0:c2}, new savings balance: {1:c2}", balance + amount, balance);

// Console.WriteLine("Old checking balance: {0:c2}, new checking balance: {1:c2}", Checking.Balance, Checking.Balance + amount);

Checking.Balance += amount;

}

else

Console.WriteLine("Insufficient balance in savings account for transfer amount of {0:c2}. Please enter smaller amount", amount);

}

else Console.WriteLine("Incorrect routing number");

}

public override string ToString()

{

return

string.Format("First Name: {0}\n", firstName) +

string.Format("Last Name: {0}\n", lastName) +

string.Format("Bank Name: {0}\n", bankName) +

string.Format("Account Type: Savings\n") +

string.Format("Routing Number: {0}\n", routingNumber) +

string.Format("Account Number: {0}\n", accountNumber) +

string.Format("Balance: {0:c2}\n", balance) +

string.Format("Interest rate: {0}%\n", interestRate \* 100);

}

~Saving() { }

}

}

// Checking class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HW09\_01

{

class Checking: BankAccount

{

// constructors

public Checking(): base()

{ }

public Checking(string first, string last, string bank, int routing, int acctNum, double bal, double apr) : base(first, last, bank, routing, acctNum, bal, apr)

{ }

public void WriteCheck(double amount)

{

Console.WriteLine("{0} {1} of account number {2}", firstName, lastName, accountNumber);

if (amount > balance)

{

double penalty = 0.1 \* amount;

balance -= penalty;

Console.WriteLine("The check for {0:c2} bounced due to insufficient fund. \nThe checking account was penalized 10% ({1:c2}) of the check",amount, penalty);

Console.WriteLine("New balance = {0:c2}", balance);

}

else

{

balance -= amount;

Console.WriteLine("Balance = {0:c2} after writing check of {1:c2}", balance, amount);

}

}

public void TransferToSaving(double amount, int routing, BankAccount Saving)

{

if (routing == Saving.RoutingNumber)

{

Console.WriteLine("{0} {1} of account number {2}", firstName, lastName, accountNumber);

if (amount <= balance)

{

balance -= amount;

Console.WriteLine("Transferred {0:c2} to savings account number {1} with routing number {2}", amount, Saving.AccountNumber, Saving.RoutingNumber);

Console.WriteLine("Old checking balance: {0:c2}, new checking balance: {1:c2}", balance + amount, balance);

Saving.Balance += amount;

}

else

Console.WriteLine("Insufficient balance in checking account for transfer amount of {0:c2}. Please enter smaller amount", amount);

}

else Console.WriteLine("Incorrect routing number");

}

public override string ToString()

{

return

string.Format("First Name: {0}\n", firstName) +

string.Format("Last Name: {0}\n", lastName) +

string.Format("Bank Name: {0}\n", bankName) +

string.Format("Account Type: Checking\n") +

string.Format("Routing Number: {0}\n", routingNumber) +

string.Format("Account Number: {0}\n", accountNumber) +

string.Format("Balance: {0:c2}\n", balance) +

string.Format("Interest rate: {0}%\n", interestRate \* 100);

}

~Checking() { }

}

}