

User

bing ai

Bing

Say goodbye to fruitless searches and hello to finding what you need with our new and improved search engine! We've enhanced Microsoft Bing's rich results with artificial intelligence for a superior experience.

User

```
using System; using System.Collections.Generic;
```

```
public interface IAccount { string FullName { get; set; } string AccountNumber { get; set; } string AccountType { get; set; } double Balance { get; set; } }
```

```
void Deposit(double amount);  
void Withdraw(double amount);
```

```
}
```

```
public class Account : IAccount { public string FullName { get; set; } public string AccountNumber { get; set; } public string AccountType { get; set; } public double Balance { get; set; } }
```

```
public void Deposit(double amount)  
{  
    Balance += amount;  
}
```

```
public void Withdraw(double amount)  
{  
    Balance -= amount;  
}
```

```
}
```

```
public class CheckingAccount : IAccount { public string FullName { get; set; } public string AccountNumber { get; set; } public string AccountType { get; set; } public double Balance { get; set; } }
```

```
public void Deposit(double amount)  
{  
    Balance += amount;  
}
```

```
public void Withdraw(double amount)  
{  
    Balance -= amount;  
}
```

```
public void Transfer(double amount, CheckingAccount toAccount)  
{  
    Withdraw(amount);  
    toAccount.Deposit(amount);  
}
```

```
}
```

```
public class SavingsAccount : IAccount { public string FullName { get; set; } public string AccountNumber { get; set; } public string AccountType { get; set; } public double Balance { get; set; } public double InterestRate { get; set; } }
```

```
public void Deposit(double amount)  
{  
    Balance += amount;  
}
```

```
public void Withdraw(double amount)  
{  
    Balance -= amount;  
}
```

```
public void CalculateInterest()  
{  
    double interest = Balance * InterestRate;  
    Deposit(interest);  
}
```

```
}
```

```
public interface IBalancePrinter { void PrintBalance(IAccount account); }
```

```
public class ConsoleBalancePrinter : IBalancePrinter { public void PrintBalance(IAccount account) { Console.WriteLine("{0} | {1} | {2} | {3:C}", account.FullName, account.AccountNumber, account.AccountType, account.Balance); } }
```

```

public interface IDisplay { void DisplayMenu(); void DisplayBalance(double balance); void DisplayTransactionHistory(List history); void
DisplayAccountInfo(IAccount account); }

public class ConsoleDisplay : IDisplay { private readonly IBalancePrinter _balancePrinter;

public ConsoleDisplay(IBalancePrinter balancePrinter) { _balancePrinter = balancePrinter; }

public void DisplayMenu() { Console.WriteLine("1. Login"); Console.WriteLine("2. Register"); Console.WriteLine("3. Exit");
Console.WriteLine("4. Print Balance"); Console.WriteLine("5. Print Account Info"); }

public void DisplayBalance(double balance) { Console.WriteLine("Your balance is {0:C}", balance); }

public void DisplayTransactionHistory(List history) { foreach (string transaction in history) { Console.WriteLine(transaction); } }

public void DisplayAccountInfo(IAccount account) { _balancePrinter.PrintBalance(account); } }

public interface IUserInput { int GetChoice(); double GetAmount(); string GetFromAccount(); string GetToAccount(); }

public class ConsoleUserInput : IUserInput { public int GetChoice() { Console.Write("Enter your choice: "); return
int.Parse(Console.ReadLine()); }

public double GetAmount() { Console.Write("Enter the amount: "); return double.Parse(Console.ReadLine()); }

public string GetFromAccount() { Console.Write("Enter the account number to transfer from: "); return Console.ReadLine(); }

public string GetToAccount() { Console.Write("Enter the account number to transfer to: "); return Console.ReadLine(); } }

public interface ITransaction { }

public class Deposit : ITransaction { }

public class Withdrawal : ITransaction { }

public class Transfer : ITransaction { }

public interface ILoginService { }

public class LoginService : ILoginService { }

public interface IRegistrationService { }

public class RegistrationService : IRegistrationService { }

```

Bing

Hello! It seems like you've pasted some code. Is there anything I can help you with?

User

yes use the to model a bank

Bing

Sure! Here's an example of how you can model a bank in C#. You can start by creating an interface for the account with properties like FullName, AccountNumber, AccountType, and Balance. Then you can create classes that implement this interface for different types of accounts like CheckingAccount and SavingsAccount. You can also create an interface for printing the balance and a class that implements this interface for printing the balance to the console.

Here's an example of how you can model a bank in C# . This example includes interfaces for the account and balance printer, classes that implement these interfaces for different types of accounts and printing the balance to the console.

```

public interface IAccount
{
    string FullName { get; set; }
    string AccountNumber { get; set; }
    string AccountType { get; set; }
    double Balance { get; set; }

    void Deposit(double amount);
    void Withdraw(double amount);
}

public class Account : IAccount
{
    public string FullName { get; set; }
    public string AccountNumber { get; set; }
    public string AccountType { get; set; }
    public double Balance { get; set; }

    public void Deposit(double amount)
    {
        Balance += amount;
    }
}

```

```

        public void Withdraw(double amount)
        {
            Balance -= amount;
        }
    }

public class CheckingAccount : IAccount
{
    public string FullName { get; set; }
    public string AccountNumber { get; set; }
    public string AccountType { get; set; }
    public double Balance { get; set; }

    public void Deposit(double amount)
    {
        Balance += amount;
    }

    public void Withdraw(double amount)
    {
        Balance -= amount;
    }

    public void Transfer(double amount, CheckingAccount toAccount)
    {
        Withdraw(amount);
        toAccount.Deposit(amount);
    }
}

public class SavingsAccount : IAccount
{
    public string FullName { get; set; }
    public string AccountNumber { get; set; }
    public string AccountType { get; set; }
    public double Balance { get; set; }
    public double InterestRate { get; set; }

    public void Deposit(double amount)
    {
        Balance += amount;
    }

    public void Withdraw(double amount)
    {
        Balance -= amount;
    }

    public void CalculateInterest()
    {
        double interest = Balance * InterestRate;
        Deposit(interest);
    }
}

public interface IBalancePrinter
{
    void PrintBalance(IAccount account);
}

public class ConsoleBalancePrinter : IBalancePrinter
{
    public void PrintBalance(IAccount account)
    {
        Console.WriteLine("{0} | {1} | {2} | {3:C}", account.FullName, account.AccountNumber, account.AccountType, account.Balance);
    }
}

```

I hope this helps!

User

use the code i pasted in a main program

Bing

Sure! Here's an example of how you can use the code you pasted in a main program:

```

using System;
using System.Collections.Generic;

namespace Bank
{
    class Program
    {
        static void Main(string[] args)
        {
            IAccount account = new Account();
            account.FullName = "John Doe";

```

```

account.AccountNumber = "1234567890";
account.AccountType = "Checking";
account.Balance = 1000;

IBalancePrinter balancePrinter = new ConsoleBalancePrinter();
balancePrinter.PrintBalance(account);

IUserInput userInput = new ConsoleUserInput();
int choice = userInput.GetChoice();

switch (choice)
{
    case 1:
        double amount = userInput.GetAmount();
        account.Deposit(amount);
        break;
    case 2:
        amount = userInput.GetAmount();
        account.Withdraw(amount);
        break;
    case 3:
        string fromAccountNumber = userInput.GetFromAccount();
        string toAccountNumber = userInput.GetToAccount();

        CheckingAccount fromAccount = new CheckingAccount();
        fromAccount.AccountNumber = fromAccountNumber;

        CheckingAccount toAccount = new CheckingAccount();
        toAccount.AccountNumber = toAccountNumber;

        amount = userInput.GetAmount();
        fromAccount.Transfer(amount, toAccount);
        break;
    default:
        Console.WriteLine("Invalid choice.");
        break;
}

balancePrinter.PrintBalance(account);
}
}
}

```

This code creates an instance of the Account class and sets its properties. It then creates instances of the ConsoleBalancePrinter and ConsoleUserInput classes and uses them to print the balance and get user input. It then uses a switch statement to perform different operations based on the user's choice. Finally, it prints the balance again.

I hope this helps!