David Jones, a department store, hired you as a Software Engineer to design and develop a **C# object-oriented solution** to determine:

* The **customers** who have exceeded the limit of $400.00 on their accounts, and
* The **suppliers** to whom David Jones owes more than $500.00

The David Jones store keeps the following information for each **customer**:

1. Customer name (must be between 6 and 15 characters)
2. Account number (must be 6 digits and start with digit ‘1’)
3. Amount owing by the customer to David Jones ($) at the beginning of a particular month (>= 0)
4. Total of all purchases ($) from David Jones by this customer for the month
5. Total of all payments ($) made by this customer to David Jones in this month; no overpayments are to be made by the customer
6. Amount owing to David Jones ($) at the end of the month

The David Jones store keeps the following information for each of its **supplier**:

1. Supplier name (must be between 5 and 16 characters)
2. Account number (must be 6 digits and start with digit ‘2’)
3. Account balance ($) at the beginning of a particular month, that is, amount owing by David Jones to this supplier ( >= 0 )
4. Total of all purchases ($) by David Jones from this supplier this month
5. Total of all payments ($) made by David Jones to this supplier this month; no overpayments are to be made by David Jones
6. Amount owing ($) at the end of the month by David Jones to this supplier

Problem analysis

The purpose of the application is to display customers and suppliers’ information, including 5 messages and then calculating the last message.

Three separate classes are developed. One is for properties of customers, one is for properties of suppliers, and last one is operation or application class.

As to customers and suppliers’ classes, there are basic constructor and properties of customer or supplier’s name, account number, month, owing number at beginning of the month, total purchase number, total payments. After that, create one method to calculate owing number at the end of the month. In addition, override tostring method for displaying above information.

As to the application class, firstly, asking whether check customer’s status. If so, then asking number of customers. Thirdly, creating an array which length is the number of customers, for each element in the array, asking above information. NB: there are some requirements in each information.

**Variables**

**Customer and Supplier (instance variables)**



Above variables are used to identify customers and suppliers’ information.

**Application (local variables)**

****

The application class that is using the customer and supplier class also needs data. It allows user to enter value for number of customers, number of suppliers, month number, customer or supplier’s name, account number, owing number at beginning of month, total purchase number, total payment number, whether check customer or supplier’s status.

**Constants**

**No constants in my application**

**Class diagrams**

**Customer and Supplier**

name: string

accountNum: int

month: int

owingNumBegin: double

purchasesNum: double

paymentsNum: double

owingNumEnd: double

GetOwingEnd(): double

ToString(): string

**Program (application)**

numOfCustomer: int

numOfSupplier: int

month: int

name: string

accountNum: string

owingNumBegin: string

purchaseNum: string

paymentNum: string

check: string

sMonth: string

Customers[] customer : array

Suppliers[] supplier : array

Check(): bool

GetNum(): void

Output(): void

GetInfo(): void

GainName(): string

GainAccountNum(): int

GainMonth(): int

OwingNumBegin(): double

PurchaseNum(): double

PaymentNum(): double

**Properties**

**Customer and supplier**

GetName **: string**

GetAccountNum **: int**

GetMonth **: int**

GetOwingBegin **: double**

GetPurchasesNum **: double**

GetPayNum **: double**

**Pseudocode**

**Customer Example**

**If(want to check customer)**

**{**

**numberOfCustomer = GetNum();**

**foreach customer**

**{**

**GetName();**

**GetAccountNum();**

**GetMonth();**

**GetOwingBegin();**

**GetPurchaseNum();**

**GetPayNum();**

**GetOwingEnd();**

**}**

**}**

**GetNum()**

**{**

**How many customers u want to check;**

**}**

**GetName()**

**{**

**While(5< name.length < 16)**

**{**

**Input customer’s name**

**Return name;**

**}**

**}**

**GetAccountNum()**

**{**

**While( number.length == 6 && number.startwith(1))**

**{**

**Input account number**

**Return account number;**

**}**

**}**

**GetMonth()**

**{**

**While(0 < number < 13)**

**{**

**Input month number;**

**Return month number;**

**}**

**}**

**GetOwingBegin()**

**{**

**While(0 < number)**

**{**

**Input Owing number to David Jones at beginning of the month;**

**Return owing number;**

**}**

**}**

**GetPurchaseNum ()**

**{**

**While(0 < number)**

**{**

**Input purchase number from David Jones at the month;**

**Return purchase number;**

**}**

**}**

**GetPayNum ()**

**{**

**While(0 < number)**

**{**

**Input payment number to David Jones at the month;**

**Return payment number;**

**}**

**}**

**GetOwingEnd()**

**{**

**Owing at end of month =** **owingBegin + purchase – payment**

**If(Owing at end of month > 400)**

**{**

**Show (**"**Credit limit exceeded.");**

**}**

**}**

**Supplier Example**

**If(want to check supplier)**

**{**

**numberOf supplier = GetNum();**

**foreach supplier**

**{**

**GetName();**

**GetAccountNum();**

**GetMonth();**

**GetOwingBegin();**

**GetPurchaseNum();**

**GetPayNum();**

**GetOwingEnd();**

**}**

**}**

**GetNum()**

**{**

**How many suppliers u want to check;**

**}**

**GetName()**

**{**

**While(5< name.length < 16)**

**{**

**Input supplier’s name**

**Return name;**

**}**

**}**

**GetAccountNum()**

**{**

**While( number.length == 6 && number.startwith(2))**

**{**

**Input account number**

**Return account number;**

**}**

**}**

**GetMonth()**

**{**

**While(0 < number < 13)**

**{**

**Input month number;**

**Return month number;**

**}**

**}**

**GetOwingBegin()**

**{**

**While(0 < number)**

**{**

**Input Owing number by David Jones at beginning of the month;**

**Return owing number;**

**}**

**}**

**GetPurchaseNum ()**

**{**

**While(0 < number)**

**{**

**Input purchase number by David Jones at the month;**

**Return purchase number;**

**}**

**}**

**GetPayNum ()**

**{**

**While(0 < number)**

**{**

**Input payment number by David Jones at the month;**

**Return payment number;**

**}**

**}**

**GetOwingEnd()**

**{**

**Owing at end of month =** **owingBegin + purchase – payment**

**If(Owing at end of month > 500)**

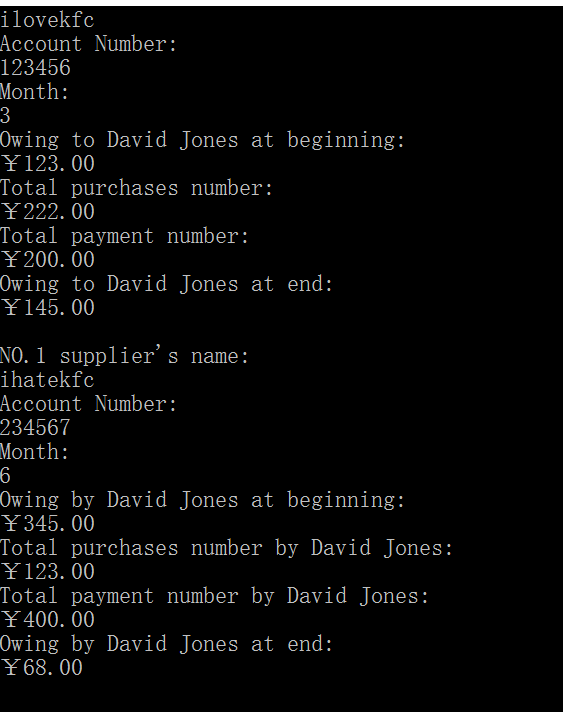
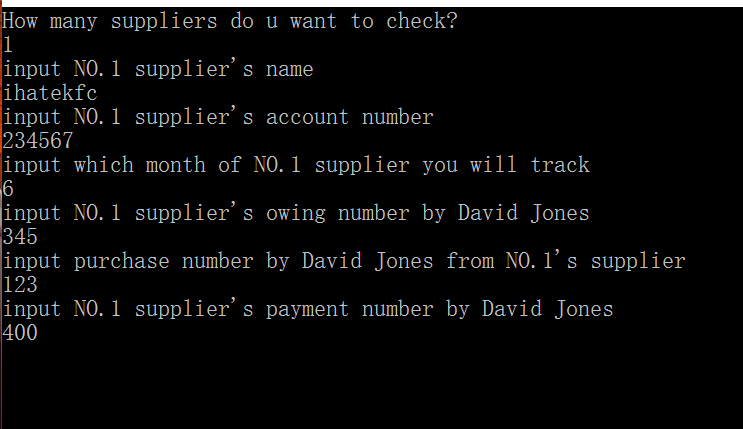
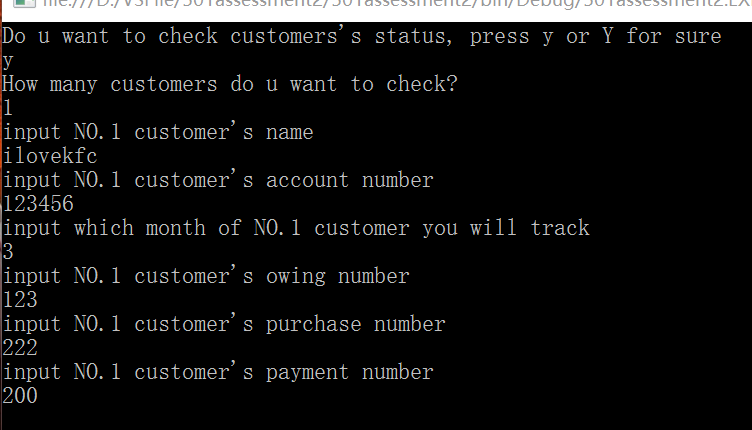
**{**

**Show (“Payment of this account is due now.");**

**}**

**}**

**Sample**

****