

Shipment –Payment Process

The payment process in the Shipping management takes a huge amount of time when it comes to a large number of customers. Because their implementation is processed one at a time so the customers are unhappy with the service provided for payment system. And so the technical team has decided to implement the payment process using multithreading to increase the performance of the system where parallel processing of payments is achieved.

The payment details are received from the user in the following format
payment id,amount,payment code

For example,

123,5000,CHEQ

1245,7888,CC

Payment code	Payment mode
CHEQ	The payment mode is by CHEQUE
CC	The payment mode is by Credit Card

Create a class named **ChequeProcessingThread** that implements the runnable interface with the following private member variables

- `ArrayBlockingQueue<String>queue`

Include appropriate getters and setters.

And include a parameterized constructor and method as follows

Access specifier	Method name	Method description
Public	<code>Void addCheque(Stringval)</code>	Add the string to the queue
Public	<code>Void run()</code>	Override the run method to take values from the queue, split the payment details string, process the payment and print the result.

Create a class named **CreditCardProcessingThread** that implements the runnable interface with the following private member variables

- `ArrayBlockingQueue<String>queue`

Include appropriate getters and setters.

And include a parameterized constructor and method as follows

Access specifier	Method	Purpose
Public	Void addCreditCard(Stringval)	Add the string to the queue
Public	Void run()	Override the run method to take values from the queue, split the payment details string, process the payment and print the result.

- Create an instance for both the classes `ChequeProcessingThread` and `CreditCardProcessingThread`
- Now create an instance of the `Thread` with the instance of `ChequeProcessingThread` as parameter and start the thread
- Similarly create an instance of the `Thread` with the instance of `CreditCardProcessingThread` as parameter and start the thread
- Read the input from the user in the main method. If the payment mode is cheque then the input is add to the `ChequeProcessingThread` thread. Similarly, if the payment mode is credit card then the input is add to the `CreditCardProcessingThread` thread
- In the run method split the string with comma and print the ouput in the following format
- `<payment mode>processing completed for payment id <payment id>`

For example

Cheque processing completed for payment id 123

Credit card processing completed for payment id 47

Hints:

Input to the thread class is send using the blocking queue which is maintained in the thread class.

A `BlockingQueue` is typically used on thread produce objects,which another thread consumes.

[Note : Strictly adhere to the object oriented specifications given as a part of the problem statement. Use the same class names,attribute names and method names.]

Input Format

The first line of input contains an integer `n`, the number of payments.

Then the payment details are received in a comma-separated string from the user in the following format **payment id,amount,payment code**

Output Format

The output contains string corresponding to each input. Please refer to the sample input and output for more details.

[All text in bold corresponds to input and rest corresponds to output.]

Sample Input and Output :

Enter the number of payment:

4

Enter all the payment details

123,5000,CHEQ

Cheque processing completed for payment id 123

1245,7888,CC

Credit card processing completed for payment id 1245

14587,23, CHEQ

Cheque processing completed for payment id 14587

478,78,CC

Credit card processing completed for payment id 47