CPAN 222 Assignment 3

# Goals

To learn how to use interfaces and abstract classes.

# Description

The accounts payable department is responsible for paying the bills of the company and making any other type of payment. Most bills are represented as a form which can be filled out to cause a cheque to be generated and sent to the company or person to whom the money is owed.

The company has two types of forms: the invoice payment and the recurring bill. An invoice is a request for payment from someone who has sold you goods or performed some service. Invoices are issued after every sale or after the services have been performed. Recurring bills are bills for fixed amounts that need to be paid on a monthly basis. Monthly rent payments would be an example of a recurring bill.

The information on an invoice payment is:

* a unique identifier for the cheque to be issued
* the name of the person or company being paid
* the amount of the payment
* the date of the payment
* the number of the invoice being paid

The information on a recurring bill is:

* a unique identifier for the cheque to be issued
* the name of the person or company being paid
* the amount of the payment
* the date of the payment
* the start date and end date of the period covered by the payment

These are called forms because they all have a method called getFormInfo() to ask the user to fill in the information required by the form. For example, the invoice will ask the user to enter the payee, amount, and the number of the invoice being paid. It will automatically generate a unique cheque number and the current date as the date of payment and fill in these values. The recurring bill will have a similar method to ask the users for the required values and store them in the instance variables.

You should create classes InvoicePayment and RecurringBillPayment to represent the information. Factor out the common information and move it to an abstract superclass called BillForm. Make sure that as much of the common data and behaviour is moved to the superclass as possible

A different type of payment is the payroll cheque. These are not created by filling in a form. The information for the payroll cheque is passed directly to the constructor for the class as the information is extracted from the separate employee record system. A PayrollPayment has the following information:

* a unique number of the cheque
* the name of the employee to pay
* the amount of the payment
* the date of the payment

To ensure that all cheque numbers are unique, the BillForm class should provide a static method to return the next unique cheque number and all classes needing cheque numbers should use this method.

Since the PayrollPayment is not a form like the other types of payment, it should not be derived from the BillForm class. Nonetheless, all of these payments share one thing in common – they must be able to generate a cheque. The way to do this is to create an interface like this:

interface ChequeGenerator {  
 void printCheque();  
}

Each of the classes will implement this interface and be able to print cheques even though they are not all in the same class hierarchy.

The last step is to create a main() class which should create an InvoicePayment and a RecurringBillPayment and use the getFormInfo() method to get the user to provide the information required. Once each class instance is complete, it should be stored in an array we will use to print the cheques. This array can be of type ChequeGenerator since all of the objects we store in the array will implement that interface. Next, create a payroll payment and add it onto the array of objects we will use to create cheques. The last step is to run through the array and print a cheque for each object in the array.

# Deliverables

You should zip your entire NetBeans project and submit it to the digital drop box on Blackboard. You should use zip on a Window PC as zip on a Mac cannot be opened on Windows and projects using rar format will not be accepted.