April 15, 2024

Assignment – 5

[100 Points]

CSC-413-02

Spring 2024

San Francisco State University

Computer Science Department

Assignment Goal:

Implementing software pattern for BankCustomer

Objective-1: [50 pts]

To understand and implement the Builder design pattern for **BankCustomer** class

Tasks:

1. Implement a Builder class named BankCustomerBuilder within the BankCustomer class to facilitate the construction of BankCustomer objects. [25 pts]

   (a) The Builder class should have setter methods for each attribute of the BankCustomer class, returning the Builder instance itself to enable method chaining

   (b) Include a build() method that constructs and returns a new BankCustomer object with the provided attributes.

2. Create a driver (test) class to test the following. [25 pts]

   Create a BankCustomer object using the Builder pattern by:

   (i) Instantiating a BankCustomerBuilder.
   (ii) Setting the attributes of the BankCustomer using the builder's setter methods
   (iii) Calling the build() method to obtain the BankCustomer object.

Submission:

Submit a Java file named BankCustomer.java containing the BankCustomer class and the BankCustomerBuilder class with appropriate implementations

Objective-2: [50 pts]

To understand and implement the Command design pattern in Java by creating a BankAccountTransaction class to perform various transactions related to bank account(s)

Tasks:

1. You should already have the BankAccount class from previous assignment with, amongst other attributes, the following attributes:

   - accountNumber (String): The account number of the bank account.
   - balance (double): The current balance of the bank account.

2. As discussed in class, Implement an interface named TransactionInterface with a method execute() to define the contract for transaction commands.
3. Implement concrete command classes that implement the Transaction interface:
   - DepositTransaction: Represents a deposit transaction.
   - WithdrawTransaction: Represents a withdrawal transaction.

4. Implement a BankAccountTransaction class that acts as an invoker in the Command pattern:
   - It should contain methods to set and execute transactions on a BankAccount object.
   - Use the Transaction interface to execute transactions
   - The BankAccountTransaction should include the following attributes at the minimum:

- id: A unique identifier for the transaction.
- datetime: The date and time when the transaction occurred.
- type: The type of transaction (e.g., deposit, withdrawal, transfer, etc.).
- amount: The amount of money involved in the transaction.
- description: A brief description or note associated with the transaction.
- source/destination Account: For transfer transactions, the source and destination accounts involved.
- status: Indicates whether the transaction was successful or failed.
- account: The user or account associated with the transaction.

5. Ensure that the BankAccount object is modified appropriately by each transaction (i.e., deposit increases the balance, withdrawal decreases the balance)

Submission:

Submit a Java file named BankAccountTransaction.java containing:

- BankAccountTransaction class
- Transaction interface
- Concrete transaction classes (DepositTransaction and WithdrawTransaction) with appropriate implementations.

Evaluation Criteria:

Your submission will be evaluated based on the following criteria:

1. Correctness and completeness of the implementation
2. Adherence to best coding practices and documentation standards as specified in the coding guidelines document
3. Accuracy of the test cases and demonstration of pattern functionality