Chau Doan
014130142

Individual Project

Credit Card Problem

Part1:
Primary Problem:

Determine the card issuer (Visa, Mater, American or add more Discover) based on the credit card number

Secondary Problems:

Create an instance of the appropriate credit card class.
Handle errors for invalid credit numbers or invalid CSV records
Design the system to accommodate adding more credit card

Design Pattern:

1. Factory Method Pattern: We'll use the Factory Method pattern to create instances of credit card class (VisaCC, MasterCC, AmExCC, DiscoverCC) based on the issuer determined by the credit card number. Following the Factory Method the subclass can decide which card can implement.
2. Strategy Pattern: We'll use the Strategy pattern to implement issuer specific validation logic for credit card numbers. Each issuer will have its validation strategy.

Consequences of Using These Pattern:

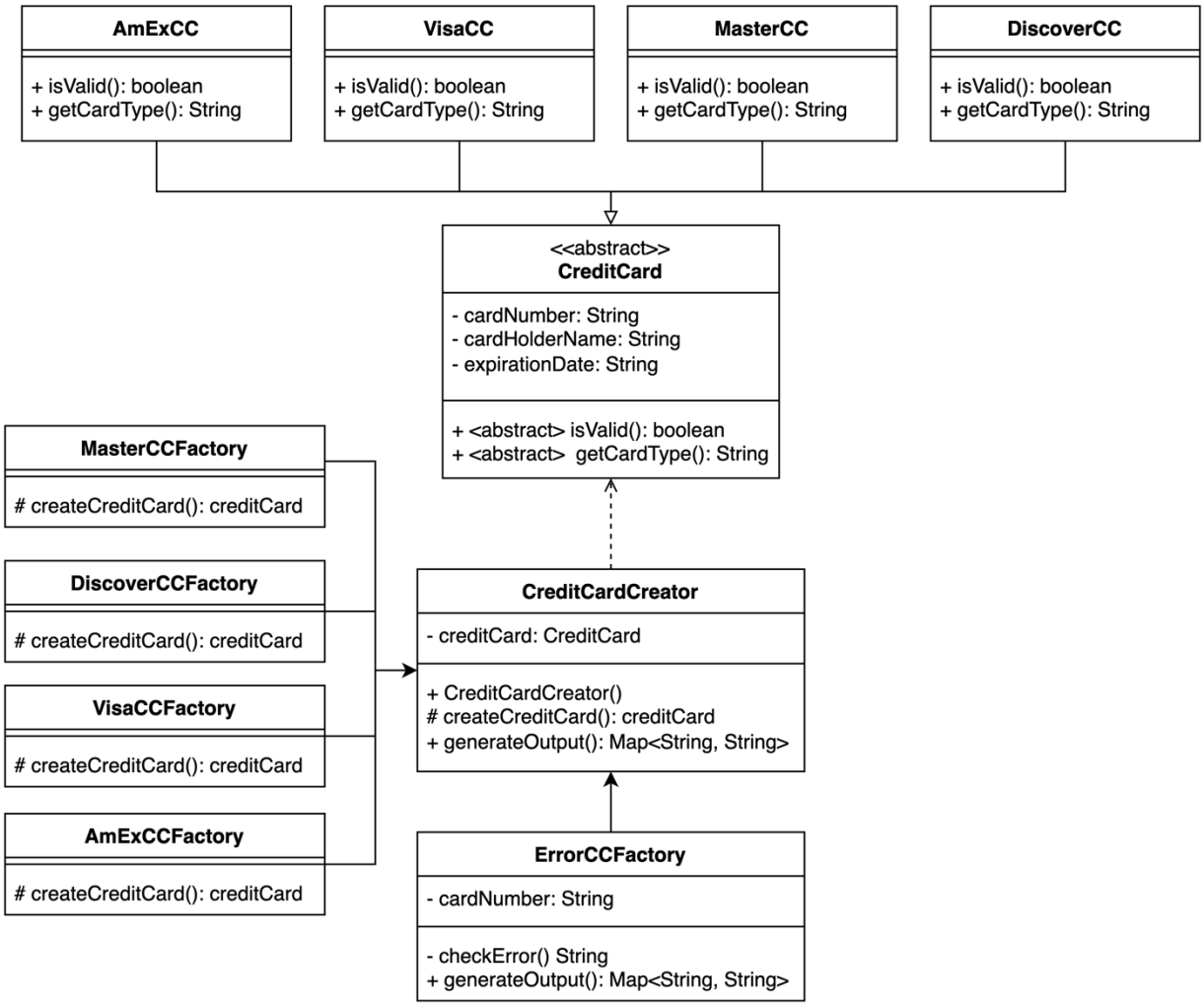
Modularity: Using the Factory Method and Strategy patterns allows we to create modular and maintainable code. Each credit card type's validation logic is encapsulated in separate classes, making it easy to add new types.

Extensibility: The design allows for easy extension by adding new credit card type. we only need to create a new concrete factory and credit card class.

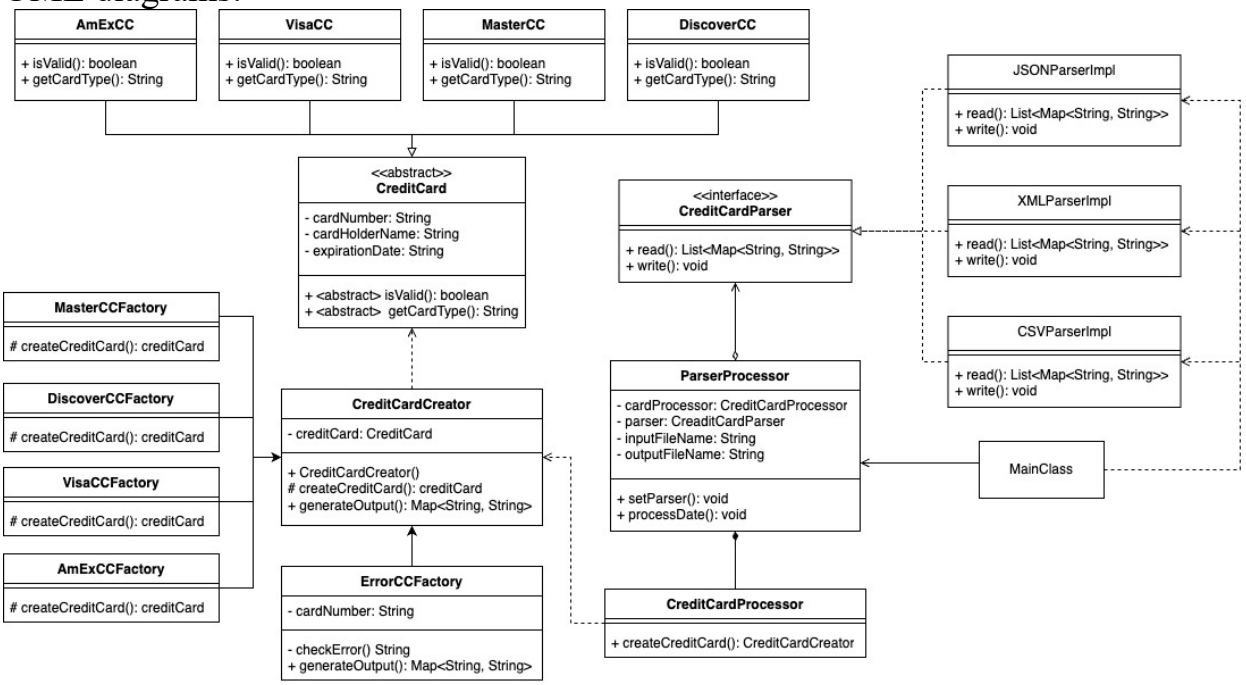
Flexibility: The Strategy pattern provides flexibility to change or extend the validation logic for existing card types without modifying the client code.

Complexity: Introducing design pattens adds some level of complexity to the code, but it is a trade-off for maintainability and extensibility.

Separation of Concerns: The Factory Method separates the creation pf objects from their use, and the Strategy pattern separates the algorithm from client, promoting a clean separation of concerns.


AmExCC
+ isValid(): boolean
+ getCardType(): String
VisaCC
+ isValid(): boolean
+ getCardType(): String
MasterCC
+ isValid(): boolean
+ getCardType(): String
DiscoverCC
+ isValid(): boolean
+ getCardType(): String
<<abstract>>
CreditCard
- cardNumber: String
- cardHolderName: String
- expirationDate: String
+ <abstract> isValid(): boolean
+ <abstract> getCardType(): String
MasterCCFactory
createCreditCard(): creditCard
DiscoverCCFactory
createCreditCard(): creditCard
VisaCCFactory
createCreditCard(): creditCard
AmExCCFactory
createCreditCard(): creditCard
CreditCardCreator
- creditCard: CreditCard
+ CreditCardCreator()
# createCreditCard(): creditCard
+ generateOutput(): Map<String, String>
ErrorCCFactory
- cardNumber: String
- checkError() String
+ generateOutput(): Map<String, String>

Part 2:
UML diagrams:



Part 3:
Coding java