Tharun Mukka

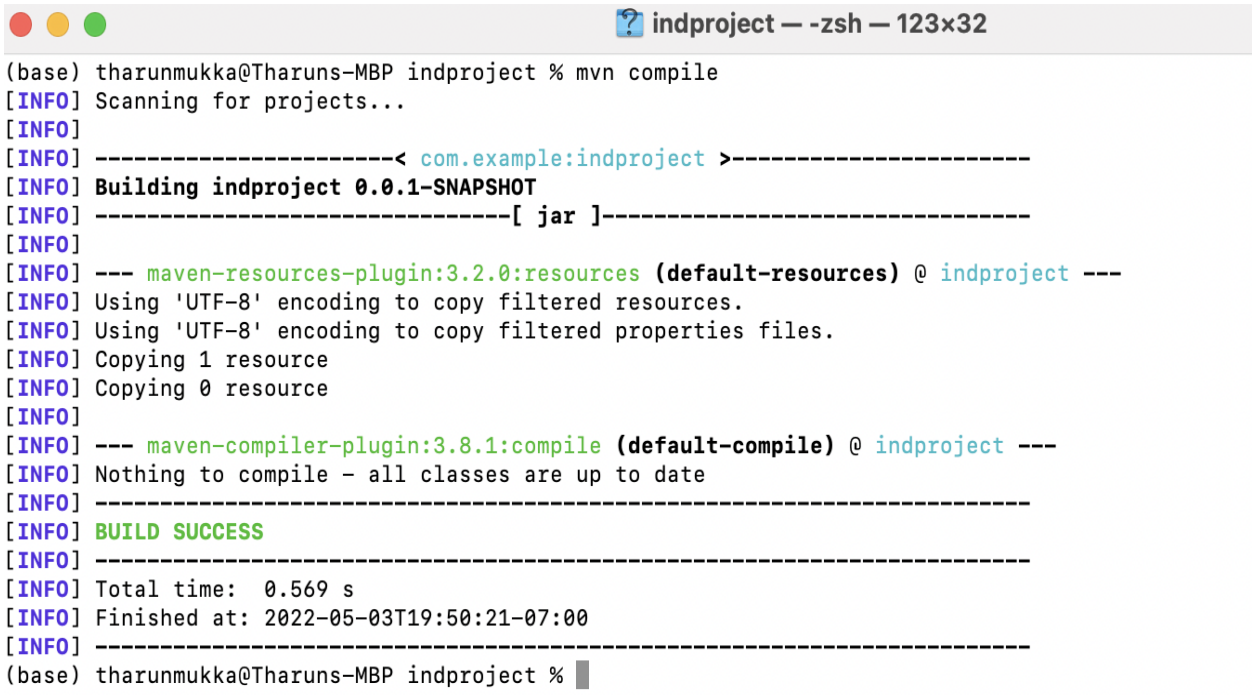
202: Individual Project

1. Detailed instructions of building the project and steps to execute the same.

Clone the repo: https://github.com/gopinathsjsu/IndividualProject_TharunMukka

Navigate to the folder where pom.xml file exists, then follow the below command one after another

- mvn compile



- mvn clean install

```
(base) tharunmukka@Tharuns-MBP indproject % mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.example:indproject >------------------------
[INFO] Building indproject 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ indproject ---
[INFO] Deleting /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ indproject ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ indproject ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 26 source files to /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target/classes
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ indproject ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ indproject ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 7 source files to /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target/test-classes
[INFO] /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/src/test/java/test/validator/CardValidatorTest.java: Some input files use or override a deprecated API
.
[INFO] /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/src/test/java/test/validator/CardValidatorTest.java: Recompile with -Xlint:deprecation for details.
[INFO]
[INFO] --- maven-surefire-plugin:2.2:test (default-test) @ indproject ---
[INFO] Surefire report directory: /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
There are no tests to run.

Results :
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ indproject ---
[INFO] Building jar: /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target/indproject-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.6.3:repackage (repackage) @ indproject ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ indproject ---
[INFO] Installing /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target/indproject-0.0.1-SNAPSHOT.jar to /Users/tharunmukka/.m2/repository/com/example/indpr
oject/0.0.1-SNAPSHOT/indproject-0.0.1-SNAPSHOT.jar
[INFO] Installing /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/pom.xml to /Users/tharunmukka/.m2/repository/com/example/indproject/0.0.1-SNAPSHOT/indproje
ct-0.0.1-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.342 s
[INFO] Finished at: 2022-05-05T10:11:25-07:00
[INFO] ------------------------------------------------------------------------
(base) tharunmukka@Tharuns-MBP indproject %
```

- Execute the below maven command to execute with arguments (Path to where the input file is located and path to where output file should be located) passed via command line
  - mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="<arg1> <arg2 > <arg3> <arg4>"
  - arg1 – path to the input data (Sample.csv)
  - arg2 – path to flight details to populate DB (flights.csv)
  - arg3 – path to Output.csv
  - arg4 – path to Output.txt

```
(base) tharunmukka@Tharuns-MBP indproject % mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="/Users/tharunmukka/Desktop/test/Sample.csv /Users/tharunmukka/Desktop/test/flights.csv /Users/tharunmukka/Deskt
op/test/Output.csv /Users/tharunmukka/Desktop/test/Output.txt"
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.example:indproject >------------------------
[INFO] Building indproject 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ indproject ---
4 params received
/Users/tharunmukka/Desktop/test/Sample.csv
/Users/tharunmukka/Desktop/test/flights.csv
/Users/tharunmukka/Desktop/test/Output.csv
/Users/tharunmukka/Desktop/test/Output.txt
Total Price for Sam is :500.0
Total Price for Richard is :1000.0
Total Price for Anna is :250.0
Total Price for Sierra is :2000.0
ticket csv file writing done!
error output file writing done!

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.7)

2022-05-05 09:07:50.751  INFO 1602 --- [unClient.main()] test.RunClient                           : Starting RunClient using Java 18.0.1 on Tharuns-MacBook-Pro.local with PID 1602 (/Users/tharunmukka/Desktop/Master
's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproject/target/classes started by tharunmukka in /Users/tharunmukka/Desktop/Master's/sem2/softwaresys/individual/IndividualProject_TharunMukka/indproj
ect)
2022-05-05 09:07:50.752  INFO 1602 --- [unClient.main()] test.RunClient                           : No active profile set, falling back to 1 default profile: "default"
2022-05-05 09:07:50.998  INFO 1602 --- [unClient.main()] test.RunClient                           : Started RunClient in 0.421 seconds (JVM running for 1.802)
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.055 s
[INFO] Finished at: 2022-05-05T09:07:51-07:00
[INFO] ------------------------------------------------------------------------
(base) tharunmukka@Tharuns-MBP indproject %
```

Sample.csv

```
BookingName, flightNumber, seatCategory, numberOfSeats, paymentCardNumber
Sam,SJ456,Economy,2,5410000000000000
Richard,BY110,Premium Economy,2,341000000000000
Anna,SJ456,Economy,1,4120000000000
John,KL908,Economy,1,6011000000000000
Sierra,BY110,Business,1,1234561323130
```

Flights.csv

```
Category(Economy,PremiumEconomy,Business),FlightNumber,AvailableSeats,Price,Arrival,Departure
Economy,SJ456,5,250,Seattle,San Jose
Premium Economy,BY110,5,500,San Francisco,New York
Business,BY110,5,2000,San Francisco,New York
Economy,CA453,5,300,Seattle,San Jose
Business,CA453,5,1500,Seattle,San Jose
```

Output.csv

```
BookingName, FlightNumber, Category, number of seats, total price
Sam,SJ456,Economy,2,500.0,
Richard,BY110,Premium Economy,2,1000.0,
Anna,SJ456,Economy,1,250.0,
```

Output.txt

```
Please enter correct booking details for John: Invalid Flight Number
Please enter correct booking details for Sierra: Invalid Card Details
```

2. Answers to the following questions:

1. Describe what is the primary problem you try to solve

   Build a flight booking application that should maintain an internal, static database (inventory of flight details) (this may be developed using HashMap's and/or other built-in Java Data structures). This means once we re-run the program, the changes to the data would not persist. We will provide the data that has to be maintained

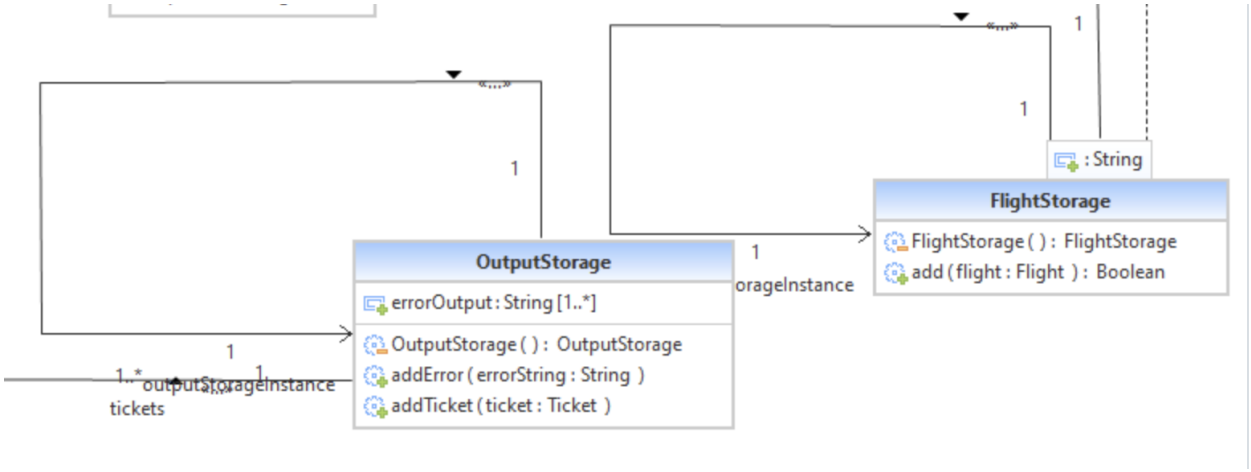
2. Describe what are the secondary problems you try to solve

By follow coding standards and best design patterns using at least three design patterns.

3. Describe what design pattern(s) you use how (use plain text and diagrams)

1. **Singleton**

Since there is only one common FlightStorage for all bookings, it's better to make the FlightStorage a singleton to make sure that only one instance of it is accessed in all the places. Similarly, OutputStorage which is used to store outputs(Tickets, errors) information is also made singleton.

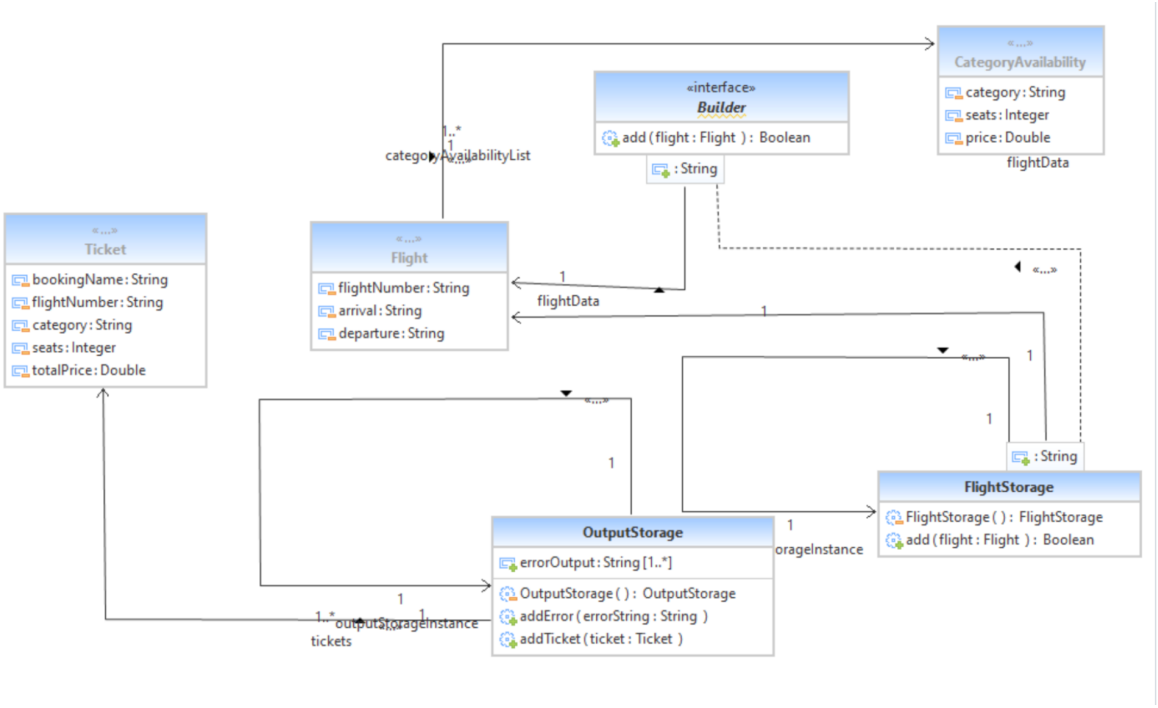


2. **Builder**

Builder is a creational design pattern that lets us construct complex objects step by step. It is used to build complex objects. The builder pattern is used to build and store data in flight storage. This involved several sub-tasks like reading data from file, creating objects of item class, and building flight database. It consists of:

Builder

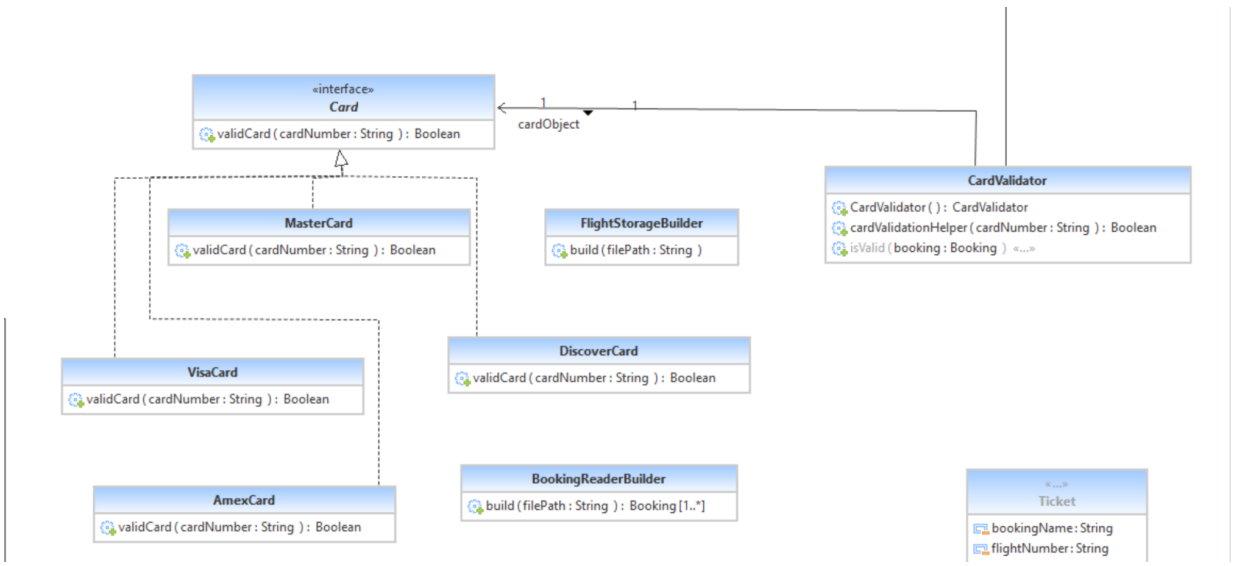
- FlightStorageBuilder
- BookingsStorageBuilder



3. **Strategy**

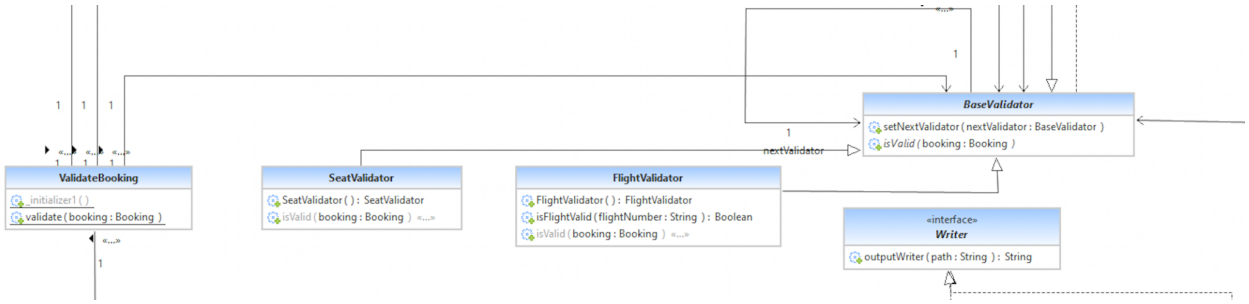
Strategy is a behavioral design pattern that lets you define a family of algorithms, put each of them into a separate class, and make their objects interchangeable.

I am creating amex, discover, visa and mastercard validation objects using straregy



4. **Chain of responsibility**

It is a behavioral design pattern that lets us pass requests along a chain of handlers. This design pattern is used to validate if the given order can be processed to give an output file. The implementation involves a main handler with abstract methods and three validation handlers implementing these abstract methods. Each handler validates the order against given conditions and decides whether to give it for further processing or reject it with a suitable error message. Validation handlers include code for validating Flight availability, seats existence in requested category and card validation.



4. Describe the consequences of using this/these pattern(s).

Singleton:

Singletons don't scale. No matter what you think should be a singleton, when your system gets bigger, it turns out you needed more than one.

Singletons are merely another way to say "global". It's not bad, but generally, it's not a good idea for systems that evolve and grow in complexity.

Strategy:

Client must be aware of different Strategies. The pattern has a potential drawback in that a client must understand how Strategies differ before it can select the appropriate one. Clients might be exposed to implementation issues.

Builder:

It does create more code (and could introduce more complexity) in the DTO than if you had for example contructor arguments and/or setters/getters.

Chain of responsibility:

The request must be received not guarantee.

The performance of the system will be affected, but also in the code debugging is not easy may cause cycle call.

5. Class diagram.

3. Junit test cases should be written for all the design pattern classes/methods. Screenshots of test execution and result should be included.