



Refactored –Assignment3 writeup

UML explanation

1. Generator

EmailGenerator is an abstract class. It depends on **InputHandler**, **Template** and **Evaluator**. It has two subclass **ActualEmailGenerator** and **OutputEmailGenerator**. They override the **handleEmail** function to send Email or output Email separately.

ActualEmailGenerator depends on **EmailParser** and **EmailSender**.

OutputEmailGenerator depends on **IOUtil** to output file.

2. Evaluator

Evaluator is an interface which can get the value for placeholders.

EvaluatorContainer is a collection of evaluators, containing all the information we can get.

KeyValueEvaluator use key-value pair to store the information.

DateEvaluator is a subclass of **KeyValueEvaluator**. It can generate current date information.

3. Info

CustomerInfo is a list of customers' data. It can use iterator to get the customer's info one by one.

FlightInfo get information from **InputHandler**.

All info will be added to evaluator to be used.

Tests explanation

I create different situations that might lead to exceptions to test Exceptions.

For some classes, for example **InputHandler**, I created specific test class for them.

For some other classes, I test them together in main class.

Other design

I didn't ask the user to add email address and email password as input argument because I think for the company, they can just set the email information once and use repeatedly.

Also, I didn't ask the user to choose whether to use **ActualEmailGenerator** or **OutputEmailGenerator** to generate Email. I think these two class can be used in different occasions. When we need the specific one, just create an instance of it.

Changes in refactored design

1. Remove the hardcode to parse the csv file.
2. Use **Evaluator** rather than **Map** to store the replacement information.
3. Remove duplicate code when handle the command line input.
4. Convert some methods with specific functions into classes.