Assignment 2: KWIC Extensions

Code Repository URL:	https://github.com/TIC3001/tic3001-ay2122s2-15		
Team number:	15		
Name:	Wang Wei	Name:	Chan Khai Wen
Student Number:	A0211395J	Student Number:	A0211493L

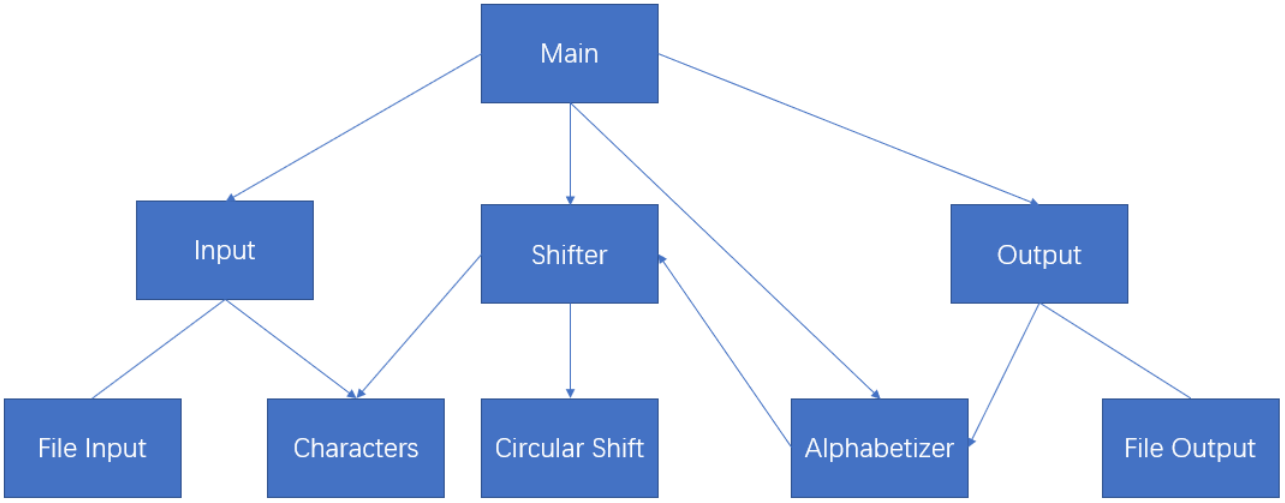
1. Introduction

In the previous assignment, we implemented Abstract Data types and Pipes and Filters. As the assignment requires extensions of functions, we choose to extend Abstract Data types design as the functions can be implemented without new modules. More elaboration can be found in Section 3. The overall flow of the program is the same as previous but with the addition of user inputs of 3 file names.

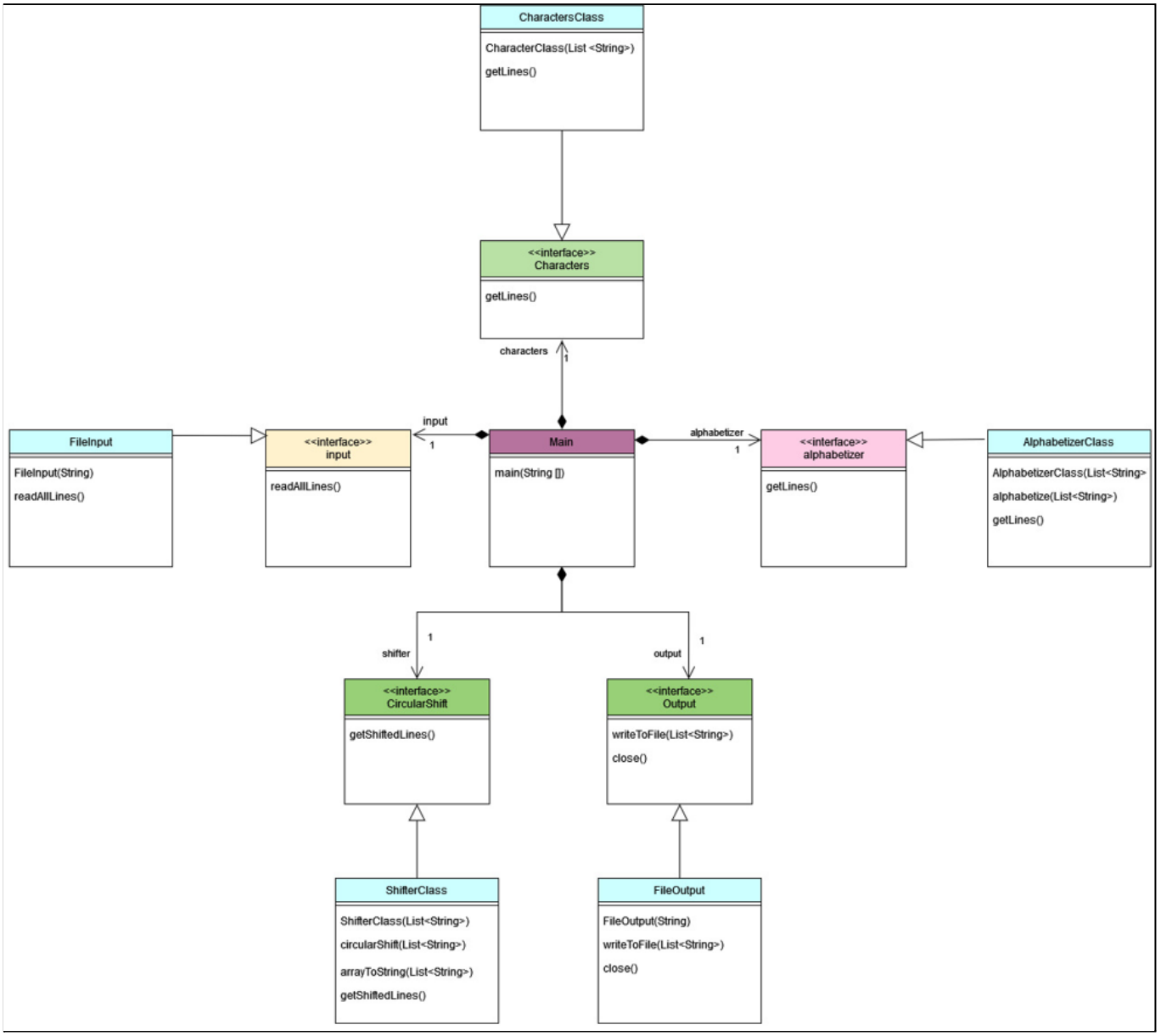
2. Architectural Design

We select Abstract Data types to be extended as the extra functions (ignore and require functions) can be handled without modification of overall architecture. It can be done through modification of a class only.

Component Diagram:



Class Diagram:



3. Applying Design Principles

We choose to implement Abstract Data type design as it has all the basic design principles such as abstraction, information hiding, Separation of Concerns (SoC), high cohesion, low coupling and Single Responsibility Principle. The operations in ADT are highly encapsulated in abstract data type/object. Each object only provides interfaces for other objects to access/communicate data (information hiding). The separation of concerns (SoC) is also implemented whereby each module addresses different types of concerns. For example, there are 5 different interfaces that handle different types of processing. The SoC also allows us to extend the functions easily in ShifterClass without affecting other modules. We just need to change the algorithms in ShifterClass to handle the extensions.

4. CI

Challenges:

When implementing Maven, we are not sure how to add the surefire add-in. After Google the result we were not able to get an exact answer. Where in the result we have to code the plug-in into the Maven file.

The current Maven version implemented if we add Junit version 5, this Maven version 2.12.4 will not be able to support it. And in result the test cases will never run and give a run result as 0.

With very little coding experience (not from Software industry and jobs not related to coding) we need longer time to process and find a solution.

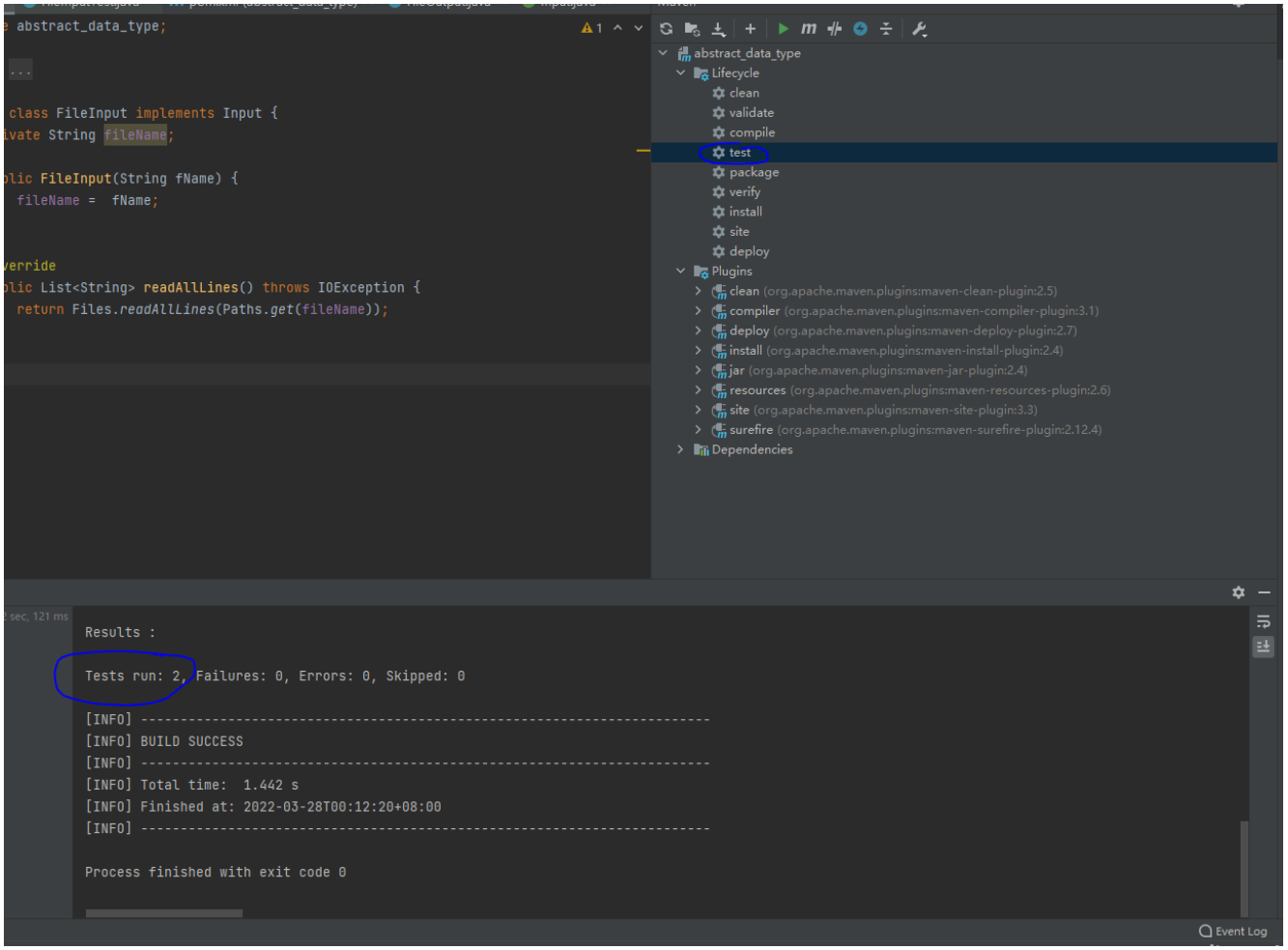
How CI helped:

We are able to create individual test cases for a specific function in the program (in our case is user enter correct/wrong input file name) where the rest of the code is not involved.

With this we can test piece by piece to find out which part of the program is not working and fix it from there.

Test cases created:

Because the user input file is the most important item in the program. The program will only run with the correct input from the user. Therefore we have made 2 test cases to test if the program will pass/fail/give error when user gives correct and incorrect file name.



6. Contributions

We choose to split the job between programming and report writing. Wang Wei was handling the programming while Khai Wen handled the report writing. We scheduled meetings during weekends to discuss our codes and the report.