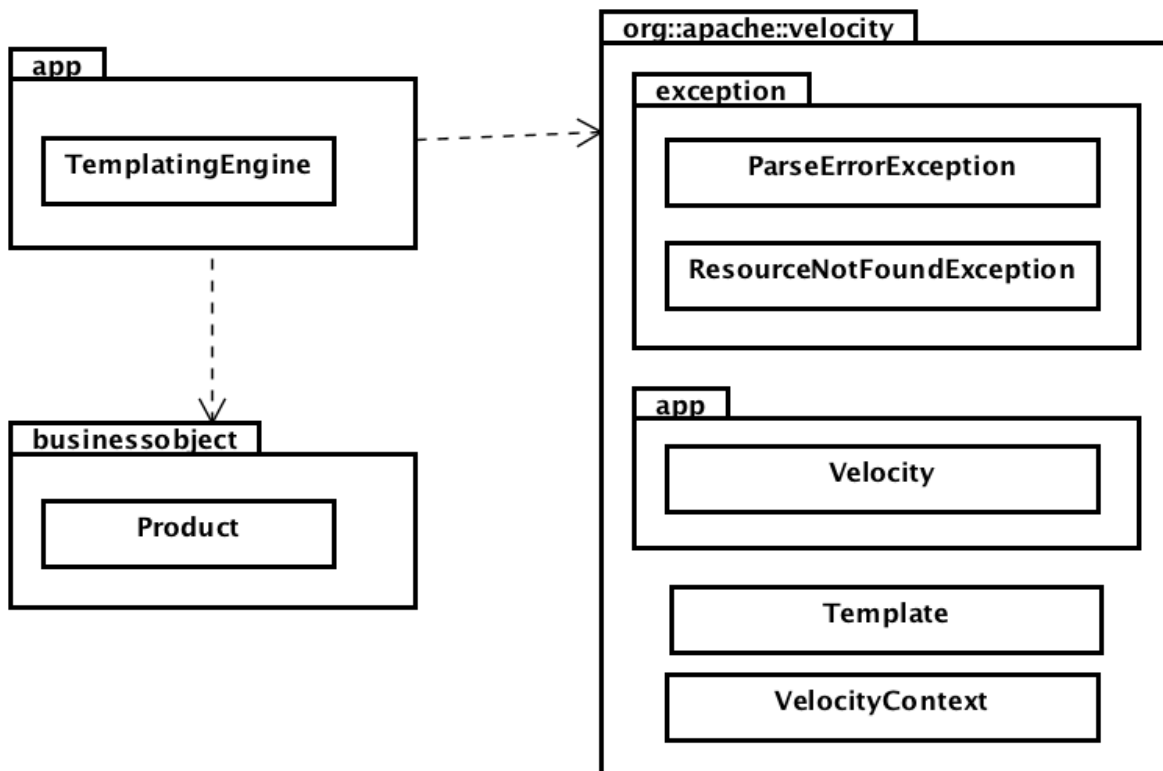


Exception

Resource: <http://blue.smu.edu.sg/exception-resource-V2.zip>

1. [**] A Templating engine allows developers to take data and insert it into a structured format, which is ubiquitous across software. The most common usage is HTML generation in web applications, or personalization of mass mails. The `TemplatingEngine` class uses a template file and data (from Java objects) to generate customized output. The following jar files are required to run the `TemplateEngine` class (`velocity-1.7.jar`, `velocity-1.7-dep.jar`). You are given the following UML class diagram



and the directory structure of the application:

```

--- Q1
  |--- template
  |   |--- example.vm
  |--- src    // directory to put your source files
  |--- lib
  |   |--- velocity-1.7.jar
  |   |--- velocity-1.7-dep.jar
  |--- classes // directory to put your project class files (exclude external libraries)
  |--- compile.bat
  |--- run.bat
  
```

Place the source Java files in the proper folders. Complete `compile.bat` and `run.bat`.

2. Simple logic with File Reading

Given a text file "numbers.txt" that contains integers, one per line.

```
10
5
-2
21
3
```

Write a program that prints the sum of all the integers read from the file.

A sample run is shown below:

```
Enter filename> a
a is invalid
Enter filename> b
b is invalid
Enter filename> numbers.txt
The sum is 37
```

3. Using delimiter method for file reading

Given a text file that contains integers, one per line.

```
10,20,30
5,9,11
-2,8,13
21,2,4
3,7,9
```

Write a program that prints the sum of the numbers of each row in the data file, to the console. A sample run is shown below:

```
Enter filename> numbers.txt
60
25
19
27
19
```

4. Reading from a file

Write a static method in InventoryApp.java called stockTaking(String fileNameAndPath) that takes in a text file (product.csv) to be read, and prints to the screen the following output:

```
There are 3 apple, each cost $1.20
There are 2 orange, each cost $0.50
There are 10 pear, each cost $0.75
There are 15 products with a total value of $12.10 to be sold.
```

This method does nothing if the filename is invalid.

5. Writing to a file

Given the Product class, complete the save method in the ProductTest class.

```
import java.util.*;

public class ProductTest {
    public static void main(String[] args){
        ArrayList<Product> products = new ArrayList<>();
        products.add(new Product("apple", 3, 1.2));
        products.add(new Product("orange", 5, 3.2));
        products.add(new Product("pear", 2, 1.1));

        save(products,"data\\product.csv");
    }

    public static void save(ArrayList<Product> productList, String pathAndFileName){
        //TODO: goes through products and saves each Product as a line in file
        //format: name,quantity,price
    }
}
```

6. * Using delimiter method for file reading

Given a text file that contains integers per line.

```
10,20,30,90
5,9
-2,8,13,2,4
21,2,4
3
```

Write a program that prints the sum of the numbers of each in the data file, to the console. A sample run is shown below:

```
Enter filename> numbers.txt
150
14
25
27
3
```

7. Given a text file that contains multiple integers per line (unfixed count of integers).

```
2,1,10,20,30,90
5,9
-2,8,13,2,4
21,2,4
3
```

Write a program that prints the numbers in each row of the data file reversed (leftmost number becomes the rightmost, vice-versa), to the console. A sample run is shown below:

```
Enter filename> numbers.txt
90,30,20,10,1,2
9,5
4,2,13,8,-2
4,2,21
3
```

Hint: You should declare an ArrayList to store the integer values. For example:

```
ArrayList<Integer> intList = new ArrayList<>();
intList.add(1); // adding the integer value 1 into the arraylist
intList.add(2);
```

Do read this website for details:

<https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html>

- END -