Task

weeJava is the programming language used by the Hobbits. **weeJava** mimics Java, and it is only a small subset of Java programming language (**see the weeJava specification document**). Examples of the weeJava programs are **HelloWeeJava.txt** and **WeeJavaExample.txt**

In this project, you need to write in **Java** a simple parser for **weeJava** programs. The parser parses an input program written in weeJava and finds its compile time errors. This is similar to what Eclipse does when you write code in Java.

Questions

There are in total five questions for this task.

Question 1 (40 marks)

To complete this project, you need to first define some help methods.

a. Create a class named **Q1** and add the **main** method

b. Create an **enum** type named **TokenType** for **weeJava** operators, symbols, keywords, identifiers, literals and Hobbits methods: OP_MULTIPLY, OP_DIVIDE, OP_MOD, OP_ADD...

```
public enum TokenType {OP_MULTIPLY, OP_DIVIDE, OP_MOD, OP_ADD, …., IDENTIFIER, STRING, …,
HOBBITS_SAY, …}
```

You should fulfil the missing part by yourself.

c. Add a method called **getOp**, which takes a parameter **ch** of type **char** and returns a value of type **TokenType**. If **ch** is one of the single char operators (e.g., +,-,*,/) in **weeJava, getOp** returns the corresponding operator name; otherwise, it returns **null**. For example, **getOp('+')** should return **TokenType.OP_ADD**.

d. Add a method (also) called **getOp**, which takes a parameter **s** of type **String** and returns a value of type **TokenType**. If **s** is one of the double char operators (e.g. >=, ==) in **weeJava**, **getOp** returns the corresponding token type; otherwise, it returns **null**. For example, **getOp("==")** should return **TokenType.OP_EQUAL**.

e. Add a method called **getSymbol**, which takes a parameter **ch** of type **char** and returns a value of type **TokenType**. If **ch** is one of the symbols (e.g., (,),[,}) in **weeJava**, **getSymbol** returns the corresponding token type; otherwise, it returns **null**.

f. Add a method called **getKeyword**, which takes a parameter **s** of type **String** and returns a value of type **TokenType**. If **s** is one of the keywords in **weeJava**, **getKeyword** returns the corresponding token type; otherwise, it returns **null**.

g. Add a method called **getHobbits**, which takes a parameter **s** of type **String** and returns a value of type **TokenType**. If **s** is one of the two Hobbits method names, **getHobbits** returns the corresponding token type; otherwise, it returns **null**.

h. Add a new method called **isLetter** that takes a parameter **ch** of type char and it returns a value of type **boolean** to indicate if the **ch** is a letter (i.e., one of the: abcdefghijklmnopqrtsuvwxyzABCDEFGHIJKLMNOPQRTSUVWXYZ). For example, **isLetter('a')** returns **true** and **isLetter('/')** returns **false**

i. Add a new method called **isDigit** that takes a parameter **ch** of type **char** and it returns a value of type **boolean** to indicate if the **ch** is a digit (i.e., one of the: 0123456789). For example, **isDigit('0')** returns **true** and **isDigit('c')** returns **false**

j. Add a new method called **isWhiteSpace** that takes a parameter **ch** of type **char** and it returns **true** if **ch** is a whitespace character; otherwise its returns **false**. Whitespace includes: **space** and **tab**. (Please make sure you understand the difference between a space and a tab)

k. Add a new method called **isLineBreak** that takes a parameter **ch** of type **char** and it returns **true** if **ch** is a new-line character; otherwise it returns **false**.

l. Edit the main method as follows. When running the **Q1** program in Eclipse, the output in the console window should look like Figure 1.

```
public static void main(String[] args) {

        TokenType op1 = getOp('+');
        System.out.println("op1: " + op1);

        TokenType op2 = getOp("==");
        System.out.println("op2: " + op2);

        TokenType sym = getSymbol('{');
        System.out.println("symbol: " + sym);

        TokenType keyword = getKeyword("int");
        System.out.println("keyword: " + keyword);
```

```
        TokenType hobbits = getHobbits("HobbitsSay");
        System.out.println("hobbits: " + hobbits);

        boolean letter = isLetter('a');
        System.out.println("letter: " + letter);

        boolean digit = isDigit('0');
        System.out.println("digit: " + digit);

        boolean whiteSpace = isWhiteSpace(' ');
        System.out.println("whiteSpace: " + whiteSpace);

        boolean newline = isLineBreak('\n');
        System.out.println("newline: " + newline);

}
```

```
op1: OP_ADD
op2: OP_EQUAL
symbol: LEFT_BRACE
keyword: KEYWORD_INT
hobbits: HOBBITS_SAY
letter: true
digit: true
whiteSpace: true
newline: true
```

Figure 1