Java Code Paranthesis Checker (100 points)

Please read important information first.

Description

In this project you are going to write a program that takes a Java code and decides if the paranthesis are correctly matched or not.

In Java codes, every opening paranthesis should match with their closing counterparts. "{" and "(" should be matched with "}" or ")", otherwise the code will not run. The only exception is if the paranthesis appear inside a string, in that case, they do not have to match.

In order to make this check, you are going to implement an algorithm using a custom stack class, called **MyStack**. This class will also be implemented by you, and the details of the implementation are given in the UML diagram later in this document.

You will implement the paranthesis checking algorithm in **ParanthesisChecker** class, **isCorrect** method. This method takes a String, javaCode, and returns a boolean, true or false. If the paranthesis are correct, your code should return true, otherwise it should return false. Do not change the method signature. Briefly, the algorithm is as follows:

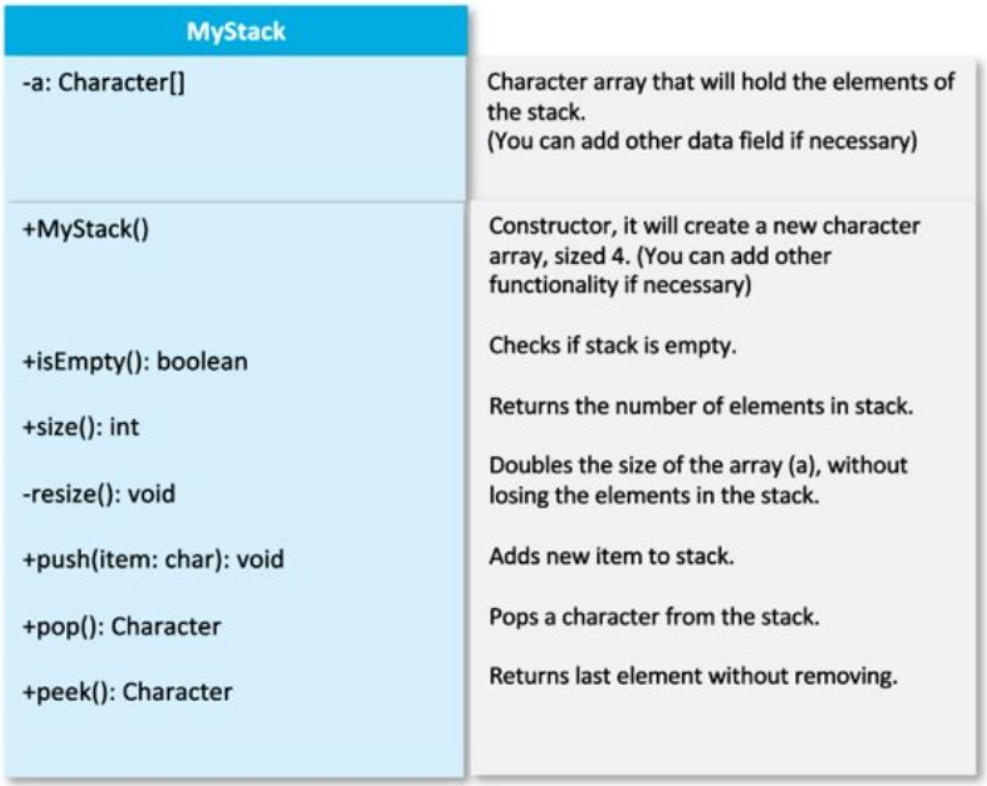
- Push an opening parenthesis onto the stack.
- If a closing parenthesis is encountered, pop a parenthesis from the stack and compare if they match. If they do not match, the source code parentheses are not correct.
- After reading the complete code, the stack should be empty. If not, parentheses are not correctly written.

MyStack Class

Stack class will be implemented by you. This class has an array of Characters as a data field. You can add new variables as data field to this class, according to your solution.

When a new stack is created, size of the array should be 4. You need to implement a private method called **resize**, to double the array size when more elements are added. For example, if you will push the fifth element to your stack, its size should be doubled before you do the operation (new size should become 8).

All methods in the following UML diagram should be implemented in MyStack class:



ParanthesisChecker Class

Complete **isCorrect** method. An example input and output are given below:

Input:

```
System.out.print("Enter a number (((even or odd): ");
int num = reader.nextInt();

if(num % 2 == 0)
    System.out.println(num + " {is even");
else
    System.out.println(num + " {is odd");
```

Output: true

Grading

- Your code will be tested with different Java codes. Remember that you will lose points if you do not implement and use **MyStack** class.

Package classification

This is the package classification described in important information.

- **Changeable**
  - question