Advanced Topics in Computer Programming ***IN - CLASS ASSESSMENT***

Prompt Description :

Read in a group of symbols and check to see if the appropriate opening symbol correctly matches up with the appropriate closing symbol.

The opening symbols are “`{(<[`“ and the appropriate closing symbols are “`})>]`“.

You must read in and analyze each group.

If you were to read in `{[]}`, you would have a correct balance of opening and closing symbols.
If you were to read in `{[}]`, you would not have a correct balance of opening and closing symbols.

Sample Data :

```
(abc(*def)
[{}]
[
[{<()>}]
{<html[value=4]*(12)>{$x}}
[one]<two>{three}(four)
car(cdr(a)(b)))
car(cdr(a)(b))
```

algorithm help

```
while there are more values in the expression
{
  get a value from the input
  if you have an opening symbol
    push it on the stack
  else if it is a close symbol
    if the stack is not empty
      pop a value
      check for a match with the current close symbol
      else
        stop the process and mark the expression as bad
}
make sure nothing is left in the stack
```

Sample Output :

```
(abc(*def) is incorrect.

[{}] is correct.

[ is incorrect.

[{<()>}] is correct.

{<html[value=4]*(12)>{$x}} is correct.

[one]<two>{three}(four) is correct.

car(cdr(a)(b))) is incorrect.

car(cdr(a)(b)) is correct.
```

Here is the constructor to get you started:

```
    public SyntaxChecker(String s) {

        exp = s;
        symbols = new Stack<String>();
    }
```

If you like, you may use the following method to split up the parameter into an array of strings:

```
    public String[] splitExpression(String expression){
        String[] list = expression.split("");
        return list;
```

You may use the testers below:

```
        SyntaxChecker test = new SyntaxChecker("(abc(*def)");
        System.out.println(test);

        SyntaxChecker test2 = new SyntaxChecker("[{}]");
        System.out.println(test2);

        SyntaxChecker test3 = new SyntaxChecker("[");
        System.out.println(test3);

        SyntaxChecker test4 = new SyntaxChecker("[{<()>}]");
        System.out.println(test4);

        SyntaxChecker test5 = new SyntaxChecker("{<html[value=4]*(12)>{$x}}");
        System.out.println(test5);

        SyntaxChecker test6 = new SyntaxChecker("[one]<two>{three}(four)");
        System.out.println(test6);

        SyntaxChecker test7 = new SyntaxChecker("car(cdr(a)(b)))");
        System.out.println(test7);
```

```
SyntaxChecker test8 = new SyntaxChecker("car(cdr(a)(b))");
System.out.println(test8);
```