Alexander Sanna

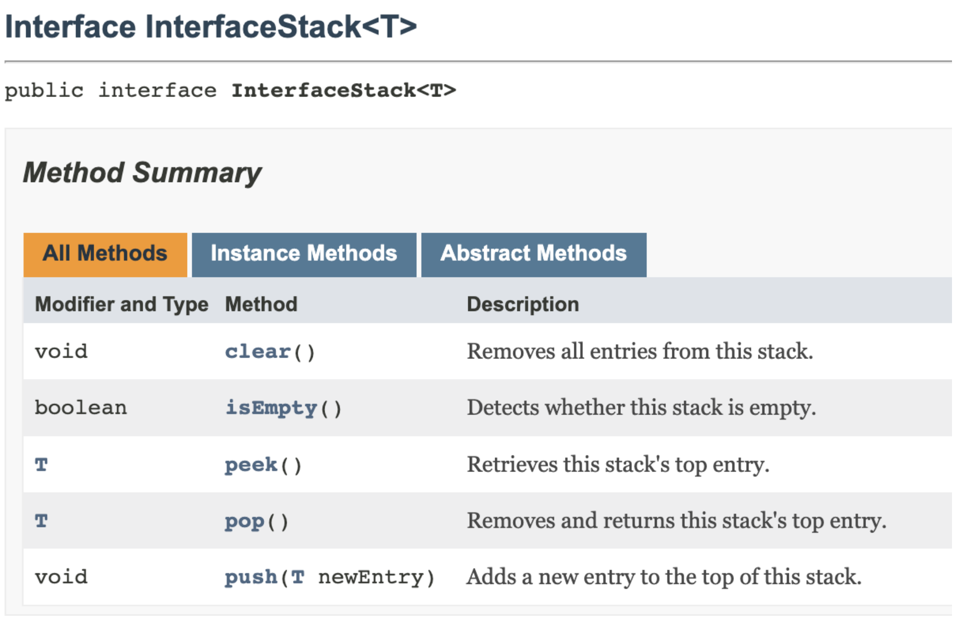
015193607

Project 3

CS 2400 Section 2

Due 10/17/22

Project Specification



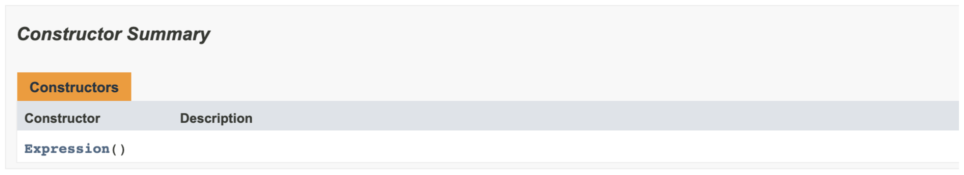
Interface InterfaceStack<T>

```
public interface InterfaceStack<T>
```

Method Summary

All Methods | Instance Methods | Abstract Methods

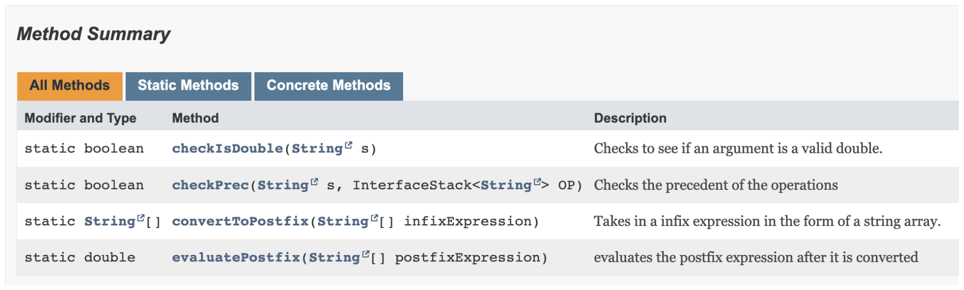
Modifier and Type	Method	Description
void	`clear()`	Removes all entries from this stack.
boolean	`isEmpty()`	Detects whether this stack is empty.
T	`peek()`	Retrieves this stack's top entry.
T	`pop()`	Removes and returns this stack's top entry.
void	`push(T newEntry)`	Adds a new entry to the top of this stack.



Constructor Summary

Constructors

Constructor	Description
`Expression()`	



Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method	Description
static boolean	`checkIsDouble(String s)`	Checks to see if an argument is a valid double.
static boolean	`checkPrec(String s, InterfaceStack<String> OP)`	Checks the precedent of the operations
static String[]	`convertToPostfix(String[] infixExpression)`	Takes in a infix expression in the form of a string array.
static double	`evaluatePostfix(String[] postfixExpression)`	evaluates the postfix expression after it is converted

For this project i utilized the Stack ADT and implemented it using an ArrayStack. This effectively solved the issues at hand regardless of command line arguments. The class ArrayStack implemented the InterfaceStack interface and the Expression class utilized the ArrayStack data type to conduct the operations.

Testing Methodology

To effectively test the error management systems in the program I inputted multiple instances of incorrect inputs to which the program responded appropriately by throwing exceptions. This made it easy to decipher what was a valid input and what would not be tolerated by the program. The command line allows for parentheses, all operations and all numerical values, and will ignore any foreign input. In the Expression class I incorporated a TestExceptions method that ran the peek and pop with an empty stack. They responded appropriately. I also included one prefixed equation:

```
"( ( 2 ^ 2 )+ 5 ) / 3 + 10 "
```

This was an expression including every operator and multiple instances of nested parentheses. It worked flawlessly.

Lessons Learned

This project taught me the effectiveness and complications when utilizing the stack adt. It was a complicated process and I got lost multiple times but now I clearly know how to use a Stack and how powerful it can be when used correctly. I also learned how to check a command line argument to see if it is a double without crashing the program by utilizing a try and catch. It was a long, tedious process but I am much more comfortable with the concepts at hand now.