Describe the code for a more readable

This is an implementation of a stack data structure in Java. The `Stack` class has three instance variables: `maxSize`, `stackArray`, and `top`.

`maxSize` is the maximum size of the stack, `stackArray` is an array used to store the elements of the stack, and `top` is an index pointing to the top element in the stack.

The class has a constructor that takes an integer `s` as an argument and initializes the `maxSize` and `stackArray` instance variables. The `top` variable is initialized to -1 to indicate an empty stack.

The `isEmpty()` method checks if the stack is empty and returns a boolean value. The `isFull()` method checks if the stack is full and returns a boolean value.

The `push(int number)` method takes an integer `number` as an argument and adds it to the top of the stack. If the stack is already full, the method prints a message saying that the stack is full.

The `pop()` method removes and returns the top element from the stack. If the stack is already empty, the method returns -99.

The `peek()` method returns the top element of the stack without removing it. If the stack is already empty, the method returns -99 and prints an error message.

Overall, this implementation provides basic functionality for a stack data structure. However, it could be improved by using exceptions instead of returning -99 to handle error conditions.