String_Inversion

Stack Class

```java
class Stack {
    private int top;
    private int capacity;
    private char[] array;

```

```java
public Stack(int capacity) {
    top = -1;
    this.capacity = capacity;
    this.array = new char[capacity];
}

```

This class represents a stack data structure using an array.

Constructor Stack(int capacity)

- Purpose: Initializes the stack with a specified capacity.
- Parameters: capacity - an integer indicating the size of the stack.
- Behavior:Sets top to -1 to indicate an empty stack.Initializes the array with the given capacity.

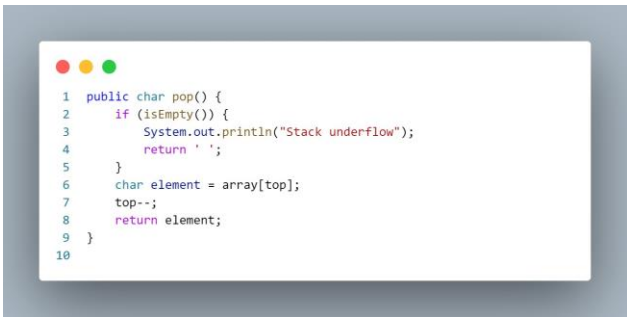
push(char element)

```java
public void push(char element) {
    if (isFull()) {
        System.out.println("Stack overflow");
        return;
    }
    top++;
    array[top] = element;
}


```

- Purpose: Pushes an element onto the stack.
- Parameters: element - the character element to be pushed onto the stack.
- Behavior:Checks if the stack is already full (isFull() method).If not full, increments top and adds the element to the array.

pop()

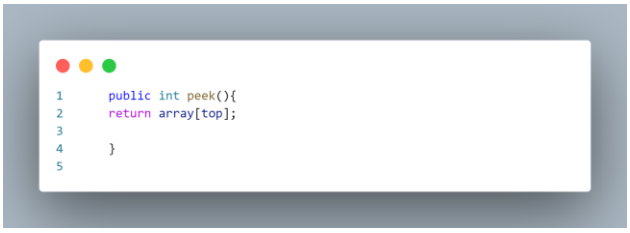


```
public char pop() {
    if (isEmpty()) {
        System.out.println("Stack underflow");
        return ' ';
    }
    char element = array[top];
    top--;
    return element;
}

```

- Purpose: Pops/removes the top element from the stack.
- Returns: The character element popped from the stack or a space character if the stack is empty.
- Behavior:Checks if the stack is empty (isEmpty() method).If not empty, retrieves the top element, decrements top, and returns the element.

peek()



```
    public int peek(){
    return array[top];

    }

```

- Purpose: Returns the element at the top of the stack without removing it.
- Returns: The character element at the top of the stack.
- Behavior:Returns the element at the top of the stack (array[top]).

isEmpty()

```
public boolean isEmpty() {
return (top == -1);
}

```

- Purpose: Checks if the stack is empty.
- Returns: true if the stack is empty, false otherwise.

isFull()

```java
public boolean isFull() {
return (top == capacity - 1);
}

```

- Purpose: Checks if the stack is full.
- Returns: true if the stack is full, false otherwise.

print()

```java
public void print(){
for(int i =0;i<=top;i++)
   System.out.println(array[i]+ " ");
System.out.println();

}

```

- Purpose: Prints all elements in the stack.
- Behavior:Loops through the elements in the stack and prints each element.

StringInverter Class

```java
class StringInverter {
    public static String invertString(String str) {
        Stack stack = new Stack(str.length());
        for (int i = 0; i < str.length(); i++) {
            stack.push(str.charAt(i));
        }

        StringBuilder invertedString = new StringBuilder();
        while (!stack.isEmpty()) {
            invertedString.append(stack.pop());
        }

        return invertedString.toString();
    }
}

```

This class provides a method to invert a given string using the Stack class.

invertString(String str)

- Purpose: Inverts a given string using a stack.
- Parameters: str - the input string to be inverted.

- Returns: The inverted string.
- Behavior:Creates an instance of the Stack class.Pushes each character of the input string onto the stack.Pops elements from the stack to construct the inverted string using StringBuilder.Returns the inverted string.

String_Inversion Class (Main)

```
1  public class String_Inversion {
2      public static void main(String[] args) {
3
4          Stack mystack = new Stack(10);
5          Scanner sc = new Scanner(System.in);
6
7          System.out.print("Enter a string: ");
8          String str = sc.next();
9
10         String invertedString = StringInverter.invertString(str);
11         System.out.println("The inverted string is: " + invertedStri
   ng);
12
13
14
15
16     }
17 }
18
```

This class contains the main method for user interaction.

main(String[] args)

- Purpose: Entry point of the program.
- Behavior:Creates an instance of Stack.Takes user input for a string.Calls invertString() from StringInverter class to get the inverted string.Prints the inverted string obtained from the invertString() method.

Overall, this program demonstrates the use of a stack to invert a given string, utilizing stack operations like push and pop to reverse the string characters.