

JOBSHEET 8

Stack



Name

Sherly Lutfi Azkiah Sulistyawati

NIM

2341720241

Class

1I

Major

Information Technology

Study Program

D4 Informatics Engineering

Practicum 1

Practice > Week8 > Stack.java > Stack > push(int)

```
1 package Week8;
2
3 public class Stack {
4     int size;
5     int top;
6     int data[];
7
8     public Stack(int size) {
9         this.size = size;
10        data = new int[size];
11        top = -1;
12    }
13
14    public boolean isEmpty() {
15        if (top == -1) {
16            return true;
17        } else {
18            return false;
19        }
20    }
21
22    public boolean IsFull() {
23        if (top == size - 1) {
24            return true;
25        } else {
26            return false;
27        }
28    }
29
30    public void push(int dt) {
31        if (!IsFull()) {
32            top++;
33            data[top] = dt;
34        } else {
35            System.out.println(x:"Stack is full");
36        }
37    }
38
39    public void pop() {
40        if (!isEmpty()) {
41            int x = data[top];
42            top--;
43            System.out.println("Remove data : " + x);
44        } else {
45            System.out.println(x:"Stack is empty");
46        }
47    }
48
49    public void peek() {
50        System.out.println("Top element : " + data[top]);
51    }
52
53    public void print() {
54        System.out.println(x:"Stack content: ");
55        for (int i = top; i >= 0; i--) {
56            System.out.println(data[i] + " ");
57        }
58        System.out.println(x:"");
59    }
60
61    public void clear() {
62        if (!isEmpty()) {
63            for (int i = top; i >= 0; i--) {
64                top--;
65            }
66            System.out.println(x:"Stack is now empty");
67        } else {
68            System.out.println(x:"Failed! Stack is still empty");
69        }
70    }
71 }
```

```
Practice > Week8 > J StackMain.java > ...
1 package Week8;
2
3 public class StackMain {
4     public static void main(String[] args) {
5
6         Stack stk = new Stack(size:5);
7         stk.push(dt:15);
8         stk.push(dt:27);
9         stk.push(dt:13);
10        stk.print();
11
12        stk.push(dt:11);
13        stk.push(dt:34);
14        stk.pop();
15        stk.peek();
16        stk.print();
17    }
18 }
```

```
Stack content:
13
27

Remove data : 34
Top element : 11
Stack content:
11
13
27
```

Question

1. In class **StackMain**, what is the usage of number 5 in this following code?

```
Stack stk = new Stack(5);
```

- In the code above, the number 5 represents the size of the stack. It's creating a stack with a capacity of 5 elements.
2. Add 2 more data in the stack with 18 and 40. Display the result!

```
Practice > Week8 > J StackMain.java > StackMain > main(String[])
1 package Week8;
2
3 public class StackMain {
4     public static void main(String[] args) {
5
6         Stack stk = new Stack(size:5);
7         stk.push(dt:15);
8         stk.push(dt:27);
9         stk.push(dt:13);
10        stk.print();
11
12        stk.push(dt:11);
13        stk.push(dt:34);
14        stk.pop();
15        stk.peek();
16        stk.print();
17
18        stk.push(dt:18);
19        stk.push(dt:40);
20        stk.print();
21    }
22 }
23 }
```

```
Stack content:
13
27

Remove data : 34
Top element : 11
Stack content:
11
13
27




Stack is full
Stack content:
18
11
13
27
```

3. In previous number, the data inserted in to the stack is only 18 and 40 is not inserted.




Why is that?

- The stack was initialized with a size of 5, so it can only hold 5 elements. When attempting to push 18 and 40, the stack was already full, so those elements couldn't be inserted. Only the elements within the size limit (15, 27, 13, 11, and 18) were successfully added to the stack.

Practicum 2

Practice > Week8 >  Book.java >  Book >  Book(String, String, int, int, int)

```
1 package Week8;
2
3 public class Book {
4     String title, authorName;
5     int publishedYear, pagesAmount, price;
6
7     public Book(String tt, String nm, int yr, int pam, int pr) {
8         title = tt;
9         authorName = nm;
10        publishedYear = yr;
11        pagesAmount = pam;
12        price = pr;
13    }
14 }
```

Practice > Week8 >  BookStack.java >  BookStack >  pop()

```
1 package Week8;
2
3 public class BookStack {
4     int size, top;
5     Book data[];
6
7     public BookStack(int size) {
8         this.size = size;
9         data = new Book[size];
10        top = -1;
11    }
12
13    public boolean IsEmpty() {
14        if (top == -1) {
15            return true;
16        } else {
17            return false;
18        }
19    }
20
21    public boolean IsFull() {
22        if (top == size - 1) {
23            return true;
24        } else {
25            return false;
26        }
27    }
28
29    public void push(Book dt) {
30        if (!IsFull()) {
31            top++;
32            data[top] = dt;
33        } else {
34            System.out.println(x: "Stack is full");
35        }
36    }
37 }
```

```

38     public void pop() {
39         if (!IsEmpty()) {
40             Book x = data[top];
41             top--;
42             System.out.println("Remove data : " + x.title + " " +
43                 x.authorName + " " + x.publishedYear + " " +
44                 x.pagesAmount + " " + x.price);
45         } else {
46             System.out.println(x:"Stack is empty");
47         }
48     }
49
50     public void peek() {
51         System.out.println("Top element : " + data[top]);
52     }
53
54     public void print() {
55         System.out.println(x:"Stack content: ");
56         for (int i = top; i >= 0; i--) {
57             System.out.println(data[i].title + " " +
58                 data[i].authorName + " " + data[i].publishedYear + " " +
59                 data[i].pagesAmount + " " + data[i].price);
60         }
61         System.out.println(x:"");
62     }
63
64     public void clear() {
65         if (!IsEmpty()) {
66             for (int i = top; i >= 0; i--) {
67                 top--;
68             }
69             System.out.println(x:"Stack is now empty");
70         } else {
71             System.out.println(x:"Failed! Stack is still empty");
72         }
73     }
74 }

```

Practice > Week8 > J BookMain.java > BookMain > main(String[])

```

1  package Week8;
2
3  import java.util.Scanner;
4
5  public class BookMain {
6      Run | Debug
7      public static void main(String[] args) {
8          BookStack st = new BookStack(size:8);
9          Scanner sc = new Scanner(System.in);
10
11          char choose;
12          do {
13              System.out.print(s:"Title : ");
14              String title = sc.nextLine();
15
16              System.out.print(s:"Author Name : ");
17              String name = sc.nextLine();
18
19              System.out.print(s:"Published Year : ");
20              int year = sc.nextInt();
21
22              System.out.print(s:"Pages Amount : ");
23              int pages = sc.nextInt();
24
25              System.out.print(s:"Price : ");
26              int price = sc.nextInt();
27
28              Book bk = new Book(title, name, year, pages, price);
29              System.out.print(s:"Do you want to add new data to stack (y/n)? ");
30              choose = sc.next().charAt(index:0);
31              System.out.println();
32              sc.nextLine();
33              st.push(bk);
34          } while (choose == 'y');
35
36          st.print();
37          st.pop();
38          st.peek();
39          st.print();

```

```

Title : Programming
Author Name : Burhantoro
Published Year : 2016
Pages Amount : 126
Price : 58000
Do you want to add new data to stack (y/n)? y

Title : Statistics
Author Name : Yasir
Published Year : 2014
Pages Amount : 98
Price : 44000
Do you want to add new data to stack (y/n)? y

Title : Economics
Author Name : Diana
Published Year : 2019
Pages Amount : 86
Price : 47500
Do you want to add new data to stack (y/n)? n

Stack content:
Economics Diana 2019 86 47500
Statistics Yasir 2014 98 44000

Remove data : Economics Diana 2019 86 47500
Top element : Week8.Book@378fd1ac
Stack content:
Statistics Yasir 2014 98 44000

```

Question

1. In class BookMain, when calling **push** method, the argument is **bk**. What information is included in the **bk** variable?
 - The argument bk is an instance of the Book class. It includes information about a specific book, such as its title, author name, published year, pages amount, and price.
2. Which of the program that its usage is to define the capacity of the stack?
 - The program that defines the capacity of the stack is BookStack. In its constructor BookStack(int size), the size parameter determines the capacity of the stack.
3. What is the function of do-while that is exist in **BookMain** class?
 - The do-while loop in the BookMain class is used to repeatedly prompt the user to input information about a book and add it to the stack until the user chooses not to add more data (choose != 'y').
4. Modify the program in **BookMain**, so that the user may choose which operation (push, pop, peek, print) to do in stack from program menu!

```

Practice > Week8 > J BookMain.java > BookMain > main(String[])
5 public class BookMain {
6     public static void main(String[] args) {
9
10         char choose;
11         do {
12             System.out.println(x:"Choose an operation:");
13             System.out.println(x:"1. Push");
14             System.out.println(x:"2. Pop");
15             System.out.println(x:"3. Peek");
16             System.out.println(x:"4. Print");
17             System.out.print(s:"Enter your choice: ");
18             int choice = sc.nextInt();
19             sc.nextLine();
20
21             switch (choice) {
22                 case 1:
23                     System.out.print(s:"Title : ");
24                     String title = sc.nextLine();
25
26                     System.out.print(s:"Author Name : ");
27                     String name = sc.nextLine();
28
29                     System.out.print(s:"Published Year : ");
30                     int year = sc.nextInt();
31
32                     System.out.print(s:"Pages Amount : ");
33                     int pages = sc.nextInt();
34
35                     System.out.print(s:"Price : ");
36                     int price = sc.nextInt();
37
38                     Book bk = new Book(title, name, year, pages, price);
39                     st.push(bk);
40                     break;
41                 case 2:
42                     st.pop();
43                     break;
44                 case 3:
45                     st.peek();
46                     break;
47                 case 4:
48                     st.print();
49                     break;
50                 default:
51                     System.out.println(x:"Invalid choice. Please choose again.");
52             }
53
54             System.out.print(s:"Back to menu (y/n)? ");
55             choose = sc.next().charAt(index:0);
56             System.out.println();
57             sc.nextLine();
58
59         } while (choose == 'y');
60

```

```

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 1
Title : Programming
Author Name : Burhantoro
Published Year : 2016
Pages Amount : 126
Price : 58000
Back to menu (y/n)? y

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 1
Title : Statistics
Author Name : Yasir
Published Year : 2014
Pages Amount : 98
Price : 44000
Back to menu (y/n)? y

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 1
Title : Economics
Author Name : Diana
Published Year : 2019
Pages Amount : 86
Price : 47500
Back to menu (y/n)? y

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 4
Stack content:
Economics Diana 2019 86 47500
Statistics Yasir 2014 98 44000

Back to menu (y/n)? y

```

```

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 2
Remove data : Economics Diana 2019 86 47500
Back to menu (y/n)? y

```

```

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 3
Top element : Week8.Book@378fd1ac
Back to menu (y/n)? y

```

```

Choose an operation:
1. Push
2. Pop
3. Peek
4. Print
Enter your choice: 4
Stack content:
Statistics Yasir 2014 98 44000

```

```

Back to menu (y/n)? n

```

```

PS D:\College\Semester 2\AlgoritmadanStrukturData>

```

Practicum 3

```
Practice > Week8 > J Postfix.java > Postfix
1  package Week8;
2
3  public class Postfix {
4      int n, top;
5      char[] stack;
6
7      public Postfix(int total) {
8          n = total;
9          top = -1;
10         stack = new char[n];
11         push(c: '(');
12     }
13
14     public void push(char c) {
15         top++;
16         stack[top] = c;
17     }
18
19     public char pop() {
20         char item = stack[top];
21         top--;
22         return item;
23     }
24
25     public boolean IsOperand(char c) {
26         if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') ||
27             (c >= '0' && c <= '9') || (c == '.' || c == ',')) {
28             return true;
29         } else {
30             return false;
31         }
32     }
33
34     public boolean IsOperator(char c) {
35         if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {
36             return true;
37         } else {
38             return false;
39         }
40     }
41
42     public int degree(char c) {
43         switch (c) {
44             case '^':
45                 return 3;
46             case '%':
47                 return 2;
48             case '/':
49                 return 2;
50             case '*':
51                 return 2;
52             case '-':
53                 return 1;
54             case '+':
55                 return 1;
56             default:
57                 return 0;
58         }
59     }
60
61     public String convert(String Q) {
62         String P = "";
63         char c;
64         for (int i = 0; i < n; i++) {
65             c = Q.charAt(i);
66             if (IsOperand(c)) {
67                 P = P + c;
68             }
69             if (c == '(') {
70                 push(c);
71             }
72             if (c == ')') {
73                 while (stack[top] != '(') {
74                     P = P + pop();
75                 }
76                 pop();
77             }
78             if (IsOperator(c)) {
79                 while (degree(stack[top]) > degree(c)) {
80                     P = P + pop();
81                 }
82                 push(c);
83             }
84         }
85         return P;
86     }
87 }
```



```
Insert mathematical expression (infix) :  
a+b*(c+d-e)/f  
Postfix : abcde-+f/*+
```

Question

1. Please explain the flow of method in **Postfix** class

- The constructor `Postfix(int total)` initializes the stack with a specified size and pushes an opening parenthesis onto the stack.
- The `push(char c)` method pushes a character onto the stack.
- The `pop()` method pops a character from the stack.
- The `IsOperand(char c)` method checks if a character is an operand (a letter, digit, space, or period).
- The `IsOperator(char c)` method checks if a character is an operator (+, -, *, /, %, ^).
- The `degree(char c)` method assigns a precedence level to each operator.
- The `convert(String Q)` method takes an infix expression Q, iterates through each character, and converts it to a postfix expression P using the stack.

2. What is the function of this program code?

```
c = Q.charAt(i);
```

- The code is assigned the character at the i-th position of the string Q. This line reads the character at the current position in the infix expression.

3. Execute the program again, how's the result if we insert **3*5^(8-6)%3** for the expression?

```
Insert mathematical expression (infix) :  
3*5^(8-6)%3  
Postfix : 3586-^3%*
```

4. In 2nd number, why the braces are not displayed in conversion result? Please explain!

- Braces are not displayed in the conversion result because postfix notation doesn't require explicit grouping; the algorithm handles operator precedence implicitly.

Assignment

1. Create a program with Stack implementation to insert a sentence and display the reversed version of the sentence as a result!

```
run:
Insert Sentence: Politeknik Negeri Malang
Result :
gnalaM iregeN kinketiloP
BUILD SUCCESSFUL (total time: 1 second)
```

```
Practice > Week8 > J ReverseSentence.java > ...
1  package Week8;
2
3  import java.util.Stack;
4
5  public class ReverseSentence {
6      public String reverseSentence(String sentence) {
7          Stack<Character> stack = new Stack<>();
8          StringBuilder reversed = new StringBuilder();
9
10         // Push each character of the sentence onto the stack
11         for (char c : sentence.toCharArray()) {
12             stack.push(c);
13         }
14
15         // Pop characters from the stack to form the reversed sentence
16         while (!stack.isEmpty()) {
17             reversed.append(stack.pop());
18         }
19
20         return reversed.toString();
21     }
22 }
23
```

```
Practice > Week8 > J ReverseSentenceMain.java > ...
1  package Week8;
2
3  import java.util.Scanner;
4
5  public class ReverseSentenceMain {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          System.out.print(s:"Insert Sentence: ");
10         String sentence = scanner.nextLine();
11
12         ReverseSentence reverse = new ReverseSentence();
13         String reversedSentence = reverse.reverseSentence(sentence);
14
15         System.out.println(x:"Result:");
16         System.out.println(reversedSentence);
17
18         scanner.close();
19     }
20 }
```

Insert Sentence: Politeknik Negeri Malang
Result:
gnalaM iregeN kinketiloP

2. Every Sunday, Dewi shops to a supermarket that is in her residential area. Everytime she finishes, she keeps the receipt of what she has bought in a wardrobe. After 2 months, She had 8 receipts. She plans to trade her 5 receipts in exchange for a voucher. Create a program using stack implementation to store Dewi's receipt. As well as the retrieving the receipts. The information that are included in a receipt are as follows:

- Transaction ID
- Date
- Quantity of items
- Total price

```
Practice > Week8 > Assignment2 > J Receipt.java > ...
1  package Week8.Assignment2;
2
3  public class Receipt {
4      private String transactionId;
5      private String date;
6      private int quantityOfItems;
7      private double totalPrice;
8
9      public Receipt(String transactionId, String date, int quantityOfItems, double totalPrice) {
10         this.transactionId = transactionId;
11         this.date = date;
12         this.quantityOfItems = quantityOfItems;
13         this.totalPrice = totalPrice;
14     }
15
16     public String toString() {
17         return "Transaction ID: " + transactionId + "\n" +
18             "Date: " + date + "\n" +
19             "Quantity of Items: " + quantityOfItems + "\n" +
20             "Total Price: $" + totalPrice;
21     }
22 }
23
```

Practice > Week8 > Assignment2 > J ReceiptStack.java > ...

```
1 package Week8.Assignment2;
2
3 public class ReceiptStack {
4     private int size;
5     private int top;
6     private Receipt[] data;
7
8     public ReceiptStack(int size) {
9         this.size = size;
10        data = new Receipt[size];
11        top = -1;
12    }
13
14    public boolean isEmpty() {
15        return top == -1;
16    }
17
18    public boolean isFull() {
19        return top == size - 1;
20    }
21
22    public void push(Receipt receipt) {
23        if (!isFull()) {
24            top++;
25            data[top] = receipt;
26        } else {
27            System.out.println(x:"Receipt stack is full.");
28        }
29    }
30
31    public Receipt pop() {
32        if (!isEmpty()) {
33            Receipt poppedReceipt = data[top];
34            top--;
35            return poppedReceipt;
36        } else {
37            System.out.println(x:"Receipt stack is empty.");
38            return null;
39        }
40    }
41 }
42
```

Practice > Week8 > Assignment2 > J ReceiptMain.java > ReceiptMain > main(String[])

```
1 package Week8.Assignment2;
2
3 import java.util.Scanner;
4
5 public class ReceiptMain {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         ReceiptStack receiptStack = new ReceiptStack(size:8); // Dewi has 8 receipts
10        char choice;
11
12        do {
13            System.out.println(x:"1. Add receipt");
14            System.out.println(x:"2. Retrieve receipt");
15            System.out.println(x:"3. Exit");
16            System.out.print(s:"Enter your choice: ");
17            int option = scanner.nextInt();
18            scanner.nextLine(); // Consume newline
19
20            switch (option) {
21                case 1:
22                    System.out.print(s:"Transaction ID: ");
23                    String transactionId = scanner.nextLine();
24
25                    System.out.print(s:"Date: ");
26                    String date = scanner.nextLine();
27
28                    System.out.print(s:"Quantity of items: ");
29                    int quantity = scanner.nextInt();
30
31                    System.out.print(s:"Total price: ");
32                    double totalPrice = scanner.nextDouble();
33
34                    Receipt receipt = new Receipt(transactionId, date, quantity, totalPrice);
35                    receiptStack.push(receipt);
36                    break;
37                case 2:
38                    if (!receiptStack.isEmpty()) {
39                        Receipt retrievedReceipt = receiptStack.pop();
40                        System.out.println(x:"Receipt retrieved:");
41                        System.out.println(retrievedReceipt);
42                    } else {
43                        System.out.println(x:"No receipts available.");
44                    }
45                case 3:
46                    break;
47            }
48        } while (choice != '3');
49    }
50 }
```

```

44         }
45         break;
46
47         case 3:
48             System.out.println(x:"Exiting program.");
49             break;
50
51         default:
52             System.out.println(x:"Invalid choice. Please enter 1, 2, or 3.");
53     }
54
55     System.out.print(s:"Do you want to continue (y/n)? ");
56     choice = scanner.next().charAt(index:0);
57     scanner.nextLine(); // Consume newline
58
59     while (Character.toLowerCase(choice) == 'y');
60
61     scanner.close();
62 }
63 }
64

```

```

1. Add receipt
2. Retrieve receipt
3. Exit
Enter your choice: 1
Transaction ID: 001
Date: 24-03-2024
Quantity of items: 3
Total price: 45000
Do you want to continue (y/n)? y
1. Add receipt
2. Retrieve receipt
3. Exit
Enter your choice: 1
Transaction ID: 002
Date: 29-05-2024
Quantity of items: 5
Total price: 75000
Do you want to continue (y/n)? y
1. Add receipt
2. Retrieve receipt
3. Exit
Enter your choice: 2
Receipt retrieved:
Transaction ID: 002
Date: 29-05-2024
Quantity of Items: 5
Total Price: $75000.0
Do you want to continue (y/n)?

```