



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 3
по курсу «Языки и методы программирования»
«Полиморфизм на основе интерфейсов в языке Java»

Студент группы ИУ9-22Б Тараканов В. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Класс, представляющий двойной стек целых чисел, с порядком на основе разности суммы элементов «левого» и «правого» стеков.

2 Результаты

Листинг 1 — Вспомогательный класс DoubleStack

```
1  import java.io.*;
2
3  class DoubleStack {
4      int size;
5      int top1, top2;
6      int arr[];
7      DoubleStack(int n)
8      {
9          arr = new int[n];
10         size = n;
11         top1 = -1;
12         top2 = size;
13     }
14
15     void push1(int x)
16     {
17
18         if (top1 < top2 - 1) {
19             top1++;
20             arr[top1] = x;
21         }
22         else {
23             System.out.println("Stack Overflow");
24             System.exit(1);
25         }
26     }
27     void push2(int x)
28     {
29
30         if (top1 < top2 - 1) {
31             top2--;
32             arr[top2] = x;
33         }
34         else {
35             System.out.println("Stack Overflow");
36             System.exit(1);
37         }
38     }
39
40 }
```

Листинг 2 — Вспомогательный класс DoubleStack(продолжение)

```
1  int pop1()  
2  {  
3      if (top1 >= 0) {  
4          int x = arr[top1];  
5          top1--;  
6          return x;  
7      }  
8      else {  
9          System.out.println("Stack Underflow");  
10         System.exit(1);  
11     }  
12     return 0;  
13 }  
14 int pop2()  
15 {  
16     if (top2 < size) {  
17         int x = arr[top2];  
18         top2++;  
19         return x;  
20     }  
21     else {  
22         System.out.println("Stack Underflow");  
23         System.exit(1);  
24     }  
25     return 0;  
26 }  
27  
28 public static void main(String args[])  
29 {  
30     DoubleStack ts = new DoubleStack(5);  
31     ts.push1(5);  
32     ts.push2(10);  
33     ts.push2(15);  
34     ts.push1(11);  
35     ts.push2(7);  
36     System.out.println("Popped element from"  
37     + " stack1 is " + ts.pop1());  
38     ts.push2(40);  
39     System.out.println("Popped element from"  
40     + " stack2 is " + ts.pop2());  
41 }  
42 }  
43
```

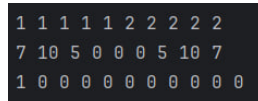
Листинг 3 — Класс sortDS, в котором реализована программа по заданию

```
1      public class sortDS implements Comparable<sortDS>{
2          private DoubleStack ds;
3          private int size;
4          public sortDS(int size , DoubleStack ds){
5              this.ds = ds;
6              this.size = size;
7          }
8          public int sum_of_array(DoubleStack ds,int left , int right){
9              int sum = 0;
10             for(int i = left;i<right;i++){
11                 sum += ds.arr[i];
12             }
13             return sum;
14         }
15         public String toString(){
16             String res = "";
17             for (int i = 0;i<this.size;i++){
18                 res += this.ds.arr[i] + " ";
19             }
20             return res;
21         }
22         public int compareTo(sortDS obj){
23             return (sum_of_array(this.ds, 0, this.ds.top1+1) - sum_of_array(
this.ds, this.ds.top2-1, this.size))-(sum_of_array(obj.ds, 0, obj.ds
.top1+1) - sum_of_array(obj.ds, obj.ds.top2-1, obj.size));
24         }
25     }
26
27
28
29
30
```

Листинг 4 — Класс Main, в котором реализована проверка работы класса sortDS

```
1  import java.util.Arrays;
2  public class Main {
3      public static void main(String[] args) {
4          DoubleStack arr1 = new DoubleStack(10);
5          DoubleStack arr2 = new DoubleStack(9);
6          DoubleStack arr3 = new DoubleStack(11);
7          arr1.push1(1);
8          arr1.push1(1);
9          arr1.push1(1);
10         arr1.push1(1);
11         arr1.push1(1);
12         arr1.push2(2);
13         arr1.push2(2);
14         arr1.push2(2);
15         arr1.push2(2);
16         arr1.push2(2);
17         arr2.push1(7);
18         arr2.push1(10);
19         arr2.push1(5);
20         arr2.push2(7);
21         arr2.push2(10);
22         arr2.push2(5);
23         arr3.push1(1);
24         sortDS[] a = new sortDS[] {
25             new sortDS(10, arr1),
26             new sortDS(9, arr2),
27             new sortDS(11, arr3)
28         };
29         Arrays.sort(a);
30         for(sortDS s: a){
31             System.out.println(s);
32         }
33     }
34 }
35
36
```

Результат запуска представлен на рисунке 1.



```
1 1 1 1 1 2 2 2 2 2
7 10 5 0 0 0 5 10 7
1 0 0 0 0 0 0 0 0 0
```

Рис. 1 — Результат