

Министерство цифрового развития, связи и массовых коммуникаций
Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Лабораторная работа № 4

«Реализация стека/дека»

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнил: студент группы БВТ1902

Лапин Виктор Андреевич

Проверил: Кутейников Иван Алексеевич

Москва

2021

Оглавление

1. ВВЕДЕНИЕ	3
2. ПОСТАНОВКА ЗАДАЧИ	3
3. ЛИСТИНГ ПРОГРАММЫ	5
Класс Stack	5
Класс Deque	6
Класс Main	7
4. ВЫВОД	22

1. ВВЕДЕНИЕ

Цель данной лабораторной работы – изучить и реализовать такие структуры данных как стек и дек, а также применить их к в решении различных задач.

Стек (stack): операции для стека: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала;

Дек (двусторонняя очередь, deque): операции для дека: инициализация, проверка на пустоту, добавление нового элемента в начало, добавление нового элемента в конец, извлечение элемента из начала, извлечение элемента из конца.

2. ПОСТАНОВКА ЗАДАЧИ

1. Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.

2. Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.

3. Даны три стержня и n дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести n дисков со стержня А на стержень С, сохранив их первоначальный порядок. При переносе дисков необходимо соблюдать следующие правила:

- на каждом шаге со стержня на стержень переносить только один диск;
- диск нельзя помещать на диск меньшего размера;
- для промежуточного хранения можно использовать стержень В.

Реализовать алгоритм, используя три стека вместо стержней А, В, С. Информация о дисках хранится в исходном файле.

4. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.

5. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек.

6. Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов.

7. Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.

8. Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

9. Дан текстовый файл. Используя стек, вычислить значение логического выражения, записанного в текстовом файле в следующей форме:

$\langle \text{ЛВ} \rangle ::= T \mid F \mid \langle \text{ЛВ} \rangle (N) \langle \text{ЛВ} \rangle \mid \langle \text{ЛВ} \rangle (A) \langle \text{ЛВ} \rangle \mid \langle \text{ЛВ} \rangle (X) \langle \text{ЛВ} \rangle \mid \langle \text{ЛВ} \rangle (O) \langle \text{ЛВ} \rangle,$

где буквами обозначены логические константы и операции:

T – True, F – False, N – Not, A – And, X – Xor, O – Or.

10. Дан текстовый файл. В текстовом файле записана формула следующего вида:

$\langle \text{Формула} \rangle ::= \langle \text{Цифра} \rangle \mid M(\langle \text{Формула} \rangle, \langle \text{Формула} \rangle) \mid N(\langle \text{Формула} \rangle, \langle \text{Формула} \rangle)$

$\langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

где буквами обозначены функции: M – определение максимума, N – определение минимума. Используя стек, вычислить значение заданного выражения.

11. Дан текстовый файл. Используя стек, проверить, является ли содержимое текстового файла правильной записью формулы вида:

$\langle \text{Формула} \rangle ::= \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle + \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle - \langle \text{Формула} \rangle$

$\langle \text{Терм} \rangle ::= \langle \text{Имя} \rangle \mid (\langle \text{Формула} \rangle)$

$\langle \text{Имя} \rangle ::= x \mid y \mid z$

3. ЛИСТИНГ ПРОГРАММЫ

Класс Stack

```
1 public class Stack {
2     private Integer[] stack;
3     private int pointerToEmpty = 0;
4
5     public Stack(int size) {
6         stack = new Integer[size];
7     }
8
9     public boolean push(Integer value) {
10         if (pointerToEmpty < stack.length) {
11             stack[pointerToEmpty] = value;
12             pointerToEmpty++;
13             return true;
14         }
15         return false;
16     }
17
18     public Integer peek() {
19         if (pointerToEmpty > 0) {
20             return stack[pointerToEmpty - 1];
21         }
22         return null;
23     }
24
25     public Integer pop() {
26         if (pointerToEmpty > 0) {
27             pointerToEmpty--;
28             return stack[pointerToEmpty];
29         }
30         return null;
31     }
32 }
```

Класс Deque

```
1 public class Deque {
2     public Integer[] deque;
3     private int size;
4     private int rightPointerToEmpty = 0;
5     private int leftPointerToEmpty = 0;
6
7     public Deque(int size) {
8         this.size = size;
9         deque = new Integer[size * 2];
10        leftPointerToEmpty = size - 1;
11        rightPointerToEmpty = size;
12    }
13
14    public boolean pushLeft(Integer value) {
15        if (leftPointerToEmpty >= 0) {
16            deque[leftPointerToEmpty] = value;
17            leftPointerToEmpty--;
18            return true;
19        }
20        return false;
21    }
22
23    public boolean pushRight(Integer value) {
24        if (rightPointerToEmpty < deque.length) {
25            deque[rightPointerToEmpty] = value;
26            rightPointerToEmpty++;
27            return true;
28        }
29        return false;
30    }
31
32    public Integer peekLeft() {
33        if (leftPointerToEmpty + 1 != rightPointerToEmpty) {
34            return deque[leftPointerToEmpty + 1];
35        }
36        return null;
37    }
38
39    public Integer peekRight() {
40        if (rightPointerToEmpty - 1 != leftPointerToEmpty) {
41            return deque[rightPointerToEmpty - 1];
42        }
43        return null;
44    }
45
46    public Integer popLeft() {
47        if (leftPointerToEmpty + 1 != rightPointerToEmpty) {
48            leftPointerToEmpty++;
49            Integer res = deque[leftPointerToEmpty];
50            if (leftPointerToEmpty + 1 == rightPointerToEmpty) {
```

```

51         leftPointerToEmpty = size - 1;
52         rightPointerToEmpty = size;
53     }
54     return res;
55 }
56 return null;
57 }
58
59 public Integer popRight() {
60     if (rightPointerToEmpty - 1 != leftPointerToEmpty) {
61         rightPointerToEmpty--;
62         Integer res = deque[rightPointerToEmpty];
63         if (rightPointerToEmpty - 1 == leftPointerToEmpty) {
64             leftPointerToEmpty = size - 1;
65             rightPointerToEmpty = size;
66         }
67         return res;
68     }
69     return null;
70 }
71 }

```

Класс Main

```

1  import java.io.*;
2  import java.util.Scanner;
3
4  public class Main {
5      public static int numHanoi = 0;
6
7      public static void main(String[] args) {
8          sortAlphabet("in1.txt", "out1.txt");
9          decrypt("in2_1.txt", "in2_2.txt", "out2.txt");
10         Hanoi("in3.txt", "out3.txt");
11         roundBalance("in4.txt", "out4.txt");
12         squareBalance("in5.txt", "out5.txt");
13         printSymbols("in6.txt", "out6.txt");
14         printNumbers("in7.txt", "out7.txt");
15         reverseStrings("in8.txt", "out8.txt");
16         logicEquation("in9.txt", "out9.txt");
17         equation2("in10.txt", "out10.txt");
18         equation3("in11.txt", "out11.txt");
19     }
20
21     public static void sortAlphabet(String inputFile,
22                                     String outputFile) {
23         System.out.println("\n\u001B[47m\u001B[30m
24         1. Сортировка строк:\u001B[0m");
25         Deque d1 = new Deque(1000);
26         Deque d2 = new Deque(1000);

```

```

25     try {
26         FileReader reader = new FileReader(inputFile);
27         int c;
28         System.out.println("Исходные строки:");
29         while ((c = reader.read()) != -1) {
30             System.out.print((char) c);
31             d2.pushRight(c);
32             if (c == 10) {
33                 if (d1.peekLeft() == null) {
34                     while (d2.peekRight() != null) {
35                         d1.pushLeft(d2.popRight());
36                     }
37                 } else {
38                     d1.pushRight(-1);
39                     while (d1.peekLeft() != -1 &&
40                         d1.peekLeft() < d2.peekLeft()) {
41                         do {
42                             d1.pushRight(d1.popLeft());
43                         } while (d1.peekRight() != -1 && d1.peekRight() != 10);
44                     }
45                     while (d1.peekLeft() != -1 &&
46                         d1.peekLeft().equals(d2.peekLeft())) {
47                         d2.pushRight(-1);
48                         int num = 0;
49                         while (!(d1.peekLeft() == -1 || d2.peekLeft() == -1 ||
50                             !d1.peekLeft().equals(d2.peekLeft()))) {
51                             num++;
52                             d1.pushRight(d1.popLeft());
53                             d2.pushRight(d2.popLeft());
54                         }
55                         if (d1.peekLeft() >= d2.peekLeft()) {
56                             while (num != 0) {
57                                 d1.pushLeft(d1.popRight());
58                                 d2.pushLeft(d2.popRight());
59                                 num--;
60                             }
61                             d2.popRight();
62                             break;
63                         } else if (d1.peekLeft() < d2.peekLeft()) {
64                             do {
65                                 d1.pushRight(d1.popLeft());
66                             } while (d1.peekRight() != -1 &&
67                                 d1.peekRight() != 10);
68                         }
69                     }
70                     while (d2.peekRight() != null) {
71                         d1.pushLeft(d2.popRight());
72                     }
73                     while (d1.peekRight() != -1) {
74                         d1.pushLeft(d1.popRight());
75                     }
76                     d1.popRight();

```



```

73         }
74     }
75 }
76 } catch (FileNotFoundException e) {
77     e.printStackTrace();
78 } catch (IOException e) {
79     e.printStackTrace();
80 }
81 try {
82     FileWriter writer = new FileWriter(outputFile, false);
83     System.out.println("\nОтсортированные строки:");
84     while (d1.peekLeft() != null) {
85         int c = d1.popLeft();
86         System.out.print((char) c);
87         writer.append((char) c);
88     }
89     writer.flush();
90 } catch (IOException e) {
91     e.printStackTrace();
92 }
93 }
94
95 public static void decrypt(String keyFile, String encryptedFile,
96                             String decryptedFile) {
97     System.out.println("\n\u001B[47m\u001B[30m
98                             2. Расшифровать сообщение:\u001B[0m");
99     Deque d = new Deque(1000);
100    String out = "";
101    try {
102        FileReader reader = new FileReader(keyFile);
103        int c;
104        System.out.println("Ключ:");
105        while ((c = reader.read()) != -1) {
106            System.out.print((char) c);
107            d.pushRight(c);
108        }
109        reader = new FileReader(encryptedFile);
110        System.out.println("\n\nЗашифрованное сообщение:");
111        while ((c = reader.read()) != -1) {
112            System.out.print((char) c);
113            int i = 0;
114            while (d.peekRight() != c && i < 28) {
115                d.pushLeft(d.popRight());
116                i++;
117            }
118            if (i == 28) {
119                out = out.concat(Character.toString(c));
120            } else {
121                d.pushLeft(d.popRight());
122                d.pushLeft(d.popRight());
123                out = out.concat(Character.toString(d.peekRight()));
124            }
125        }
126    }

```

```

123     }
124 } catch (FileNotFoundException e) {
125     e.printStackTrace();
126 } catch (IOException e) {
127     e.printStackTrace();
128 }
129 System.out.println("\n\nРасшифрованное сообщение:");
130 System.out.println(out);
131 try {
132     FileWriter writer = new FileWriter(decryptedFile, false);
133     writer.write(out);
134     writer.flush();
135 } catch (IOException e) {
136     e.printStackTrace();
137 }
138 }
139
140 public static void Hanoi(String inputFile, String outputFile) {
141     int c, col;
142     System.out.println("\n\u001B[47m\u001B[30m
143                                     3. Ханойская башня:\u001B[0m");
144     Stack s1, s2, s3 = null;
145     try {
146         FileReader reader = new FileReader(inputFile);
147         col = reader.read() - 48;
148         System.out.println("Количество дисков: " + col);
149         s1 = new Stack(col);
150         s2 = new Stack(col);
151         s3 = new Stack(col);
152         for (int i = col; i > 0; i--) {
153             s1.push(i);
154         }
155         HanoiSolver(col, s1, s2, s3);
156     } catch (FileNotFoundException e) {
157         e.printStackTrace();
158     } catch (IOException e) {
159         e.printStackTrace();
160     }
161     try {
162         FileWriter writer = new FileWriter(outputFile, false);
163         System.out.println("Решено за " + numHanoi + " ходов");
164         writer.write("Решено за " + numHanoi + " ходов");
165         writer.flush();
166     } catch (IOException e) {
167         e.printStackTrace();
168     }
169 }
170 public static void HanoiSolver(int q, Stack from, Stack buf,
171                               Stack to) {
172     if (q == 0) {
173         return;

```

```

173     }
174     numHanoi++;
175     HanoiSolver(q - 1, from, to, buf);
176     to.push(from.pop());
177     HanoiSolver(q - 1, buf, from, to);
178 }
179
180 public static void roundBalance(String inputFile,
                                String outputFile) {
181     System.out.println("\n\u001B[47m\u001B[30m4. Проверить баланс
                                круглых скобок:\u001B[0m");
182     boolean b = true;
183     Stack s = new Stack(100);
184     try {
185         FileReader reader = new FileReader(inputFile);
186         int c;
187         System.out.println("Исходный файл: ");
188         while ((c = reader.read()) != -1) {
189             System.out.print((char) c);
190             if (c == 40) {
191                 s.push(1);
192             }
193             if (c == 41) {
194                 if (s.peek() != null) {
195                     s.pop();
196                 } else {
197                     b = false;
198                     break;
199                 }
200             }
201         }
202         if (s.peek() != null) {
203             b = false;
204         }
205     } catch (FileNotFoundException e) {
206         e.printStackTrace();
207     } catch (IOException e) {
208         e.printStackTrace();
209     }
210     String res;
211     try {
212         System.out.println("");
213         FileWriter writer = new FileWriter(outputFile, false);
214         if (b) {
215             res = "Баланс круглых скобок соблюдается";
216         } else {
217             res = "Баланс круглых скобок нарушен";
218         }
219         writer.write(res);
220         System.out.println(res);
221         writer.flush();
222     } catch (IOException e) {

```

```

223     e.printStackTrace();
224 }
225 }
226
227 public static void squareBalance(String inputFile,
                                   String outputFile) {
228     System.out.println("\n\u001B[47m\u001B[30m5. Проверить баланс
                                   квадратных скобок:\u001B[0m");
229     boolean b = true;
230     Deque s = new Deque(100);
231     try {
232         FileReader reader = new FileReader(inputFile);
233         int c;
234         System.out.println("Исходный файл: ");
235         while ((c = reader.read()) != -1) {
236             System.out.print((char) c);
237             if (c == 40) {
238                 s.pushRight(1);
239             }
240             if (c == 41) {
241                 if (s.peekRight() != null) {
242                     s.popRight();
243                 } else {
244                     b = false;
245                     break;
246                 }
247             }
248         }
249         if (s.peekRight() != null) {
250             b = false;
251         }
252     } catch (FileNotFoundException e) {
253         e.printStackTrace();
254     } catch (IOException e) {
255         e.printStackTrace();
256     }
257     String res;
258     try {
259         System.out.println("");
260         FileWriter writer = new FileWriter(outputFile, false);
261         if (b) {
262             res = "Баланс квадратных скобок соблюдается";
263         } else {
264             res = "Баланс квадратных скобок нарушен";
265         }
266         writer.write(res);
267         System.out.println(res);
268         writer.flush();
269     } catch (IOException e) {
270         e.printStackTrace();
271     }
272 }

```

```

273
274 public static void printSymbols(String inputFile,
                                String outputFile) {
275     System.out.println("\n\u001B[47m\u001B[30m6. Вывести сначала
                                цифры, затем буквы, затем символы:\u001B[0m");
276     Stack numbers = new Stack(1000);
277     Stack letters = new Stack(1000);
278     Stack symbols = new Stack(1000);
279     Stack reverse = new Stack(1000);
280     Integer c;
281     try {
282         FileReader reader = new FileReader(inputFile);
283         System.out.println("Исходный файл: ");
284         while ((c = reader.read()) != -1) {
285             System.out.print((char)(int) c);
286             if (c > 47 && c < 58) {
287                 numbers.push(c);
288             } else if (c > 64 && c < 91 || c > 96 && c < 123 || c > 1039
                        && c < 1104 || c == 1105 || c == 1025) {
289                 letters.push(c);
290             } else {
291                 symbols.push(c);
292             }
293         }
294     } catch (FileNotFoundException e) {
295         e.printStackTrace();
296     } catch (IOException e) {
297         e.printStackTrace();
298     }
299     try {
300         FileWriter writer = new FileWriter(outputFile, false);
301         System.out.println("\n\nРезультат: ");
302         while ((c = numbers.pop()) != null) {
303             reverse.push(c);
304         }
305         while ((c = reverse.pop()) != null) {
306             System.out.print((char)(int) c);
307             writer.append((char)(int) c);
308         }
309         while ((c = letters.pop()) != null) {
310             reverse.push(c);
311         }
312         while ((c = reverse.pop()) != null) {
313             System.out.print((char)(int) c);
314             writer.append((char)(int) c);
315         }
316         while ((c = symbols.pop()) != null) {
317             reverse.push(c);
318         }
319         while ((c = reverse.pop()) != null) {
320             System.out.print((char)(int) c);
321             writer.append((char)(int) c);

```

```

322     }
323     writer.flush();
324 } catch (IOException e) {
325     e.printStackTrace();
326 }
327 System.out.print("\n");
328 }
329
330 public static void printNumbers(String inputFile,
                                String outputFile) {
331     System.out.println("\n\u001B[47m\u001B[30m7. Вывести сначала
                                положительные, затем отрицательные
                                числа:\u001B[0m");
332     Deque pos = new Deque(1000);
333     Deque neg = new Deque(1000);
334     Integer c;
335     try {
336         Scanner sc = new Scanner(new File(inputFile));
337         System.out.println("Исходный файл: ");
338         while (sc.hasNext()) {
339             if (sc.hasNextInt()) {
340                 c = sc.nextInt();
341                 System.out.print(c + " ");
342                 if (c < 0) {
343                     neg.pushLeft(c);
344                 } else {
345                     pos.pushLeft(c);
346                 }
347             } else {
348                 System.out.print(sc.next());
349             }
350         }
351     } catch (FileNotFoundException e) {
352         e.printStackTrace();
353     }
354     try {
355         FileWriter writer = new FileWriter(outputFile, false);
356         System.out.println("\n\nРезультат: ");
357         while ((c = neg.popRight()) != null) {
358             System.out.print((int) c + " ");
359             writer.append(String.valueOf(c));
360             writer.append(" ");
361         }
362         while ((c = pos.popRight()) != null) {
363             System.out.print((int) c + " ");
364             writer.append(String.valueOf(c));
365             writer.append(" ");
366         }
367         writer.flush();
368     } catch (IOException e) {
369         e.printStackTrace();
370     }

```

```

371     System.out.println("");
372 }
373
374 public static void reverseStrings(String inputFile,
                                   String outputFile) {
375     System.out.println("\n\u001B[47m\u001B[30m8. Вывести строки в
                                   обратном порядке:\u001B[0m");
376     Stack s1 = new Stack(1000);
377     Stack s2 = new Stack(1000);
378     int c;
379     try {
380         FileReader reader = new FileReader(inputFile);
381         System.out.println("Исходные строки: ");
382         while ((c = reader.read()) != -1) {
383             System.out.print((char) c);
384             s1.push(c);
385         }
386     } catch (FileNotFoundException e) {
387         e.printStackTrace();
388     } catch (IOException e) {
389         e.printStackTrace();
390     }
391     try {
392         FileWriter writer = new FileWriter(outputFile, false);
393         System.out.println("\nРезультат: ");
394         while (s1.peek() != null) {
395             do {
396                 s2.push(s1.pop());
397             } while (s1.peek() != null && s1.peek() != 10);
398             while (s2.peek() != null) {
399                 c = s2.pop();
400                 System.out.print((char) c);
401                 writer.append((char) c);
402             }
403         }
404         writer.flush();
405     } catch (IOException e) {
406         e.printStackTrace();
407     }
408 }
409
410 public static void logicEquation(String inputFile,
                                   String outputFile) {
411     System.out.println("\n\u001B[47m\u001B[30m9. Вычислить значение
                                   логического выражения:\u001B[0m");
412     String res = "";
413     String str = "";
414     Stack s1 = new Stack(1000);
415     int c;
416     try {
417         FileReader reader = new FileReader(inputFile);
418         System.out.println("Выражение: ");

```

```

419 while ((c = reader.read()) != -1) {
420     System.out.print((char) c);
421     if (c == 'T' || c == 'F') {
422         str = str.concat(String.valueOf((char) c));
423     }
424     if (c == 'N' || c == '(') {
425         s1.push(c);
426     }
427     if (c == ')') {
428         while (s1.peek() != null && s1.peek() != 40) {
429             str = str.concat(String.valueOf((char)(int) s1.pop()));
430         }
431         if (s1.peek() == null) {
432             res = "Ошибка в выражении";
433             break;
434         }
435         if (s1.peek() == '(') {
436             s1.pop();
437         }
438     }
439     if (c == 'A' || c == 'X' || c == 'O') {
440         while (s1.peek() != null && (s1.peek() == 'N' ||
441             s1.peek() == 'A' || s1.peek() == 'X' ||
442             s1.peek() == 'O')) {
443             str = str.concat(String.valueOf((char)(int) s1.pop()));
444         }
445         s1.push(c);
446     }
447     while (s1.peek() != null) {
448         if (s1.peek() == '(') {
449             res = "Ошибка в выражении";
450             break;
451         }
452         str = str.concat(String.valueOf((char)(int) s1.pop()));
453     }
454     for (int i = 0; i < str.length(); i++) {
455         if (str.charAt(i) == 'T' || str.charAt(i) == 'F') {
456             s1.push((int) str.charAt(i));
457         }
458         if (str.charAt(i) == 'N') {
459             if (s1.pop() == 'T') {
460                 s1.push((int)
461                     'F');
462             } else {
463                 s1.push((int)
464                     'T');
465             }
466         }
467         if (str.charAt(i) == 'A') {
468             boolean b1, b2;
469             b1 = s1.pop() == 'T';

```



```

469         b2 = s1.pop() == 'T';
470         s1.push(b1 && b2 ? (int)
471             'T' : (int)
472             'F');
473     }
474     if (str.charAt(i) == 'X') {
475         boolean b1, b2;
476         b1 = s1.pop() == 'T';
477         b2 = s1.pop() == 'T';
478         s1.push(b1 & b2 ? (int)
479             'T' : (int)
480             'F');
481     }
482     if (str.charAt(i) == 'O') {
483         boolean b1, b2;
484         b1 = s1.pop() == 'T';
485         b2 = s1.pop() == 'T';
486         s1.push(b1 || b2 ? (int)
487             'T' : (int)
488             'F');
489     }
490 }
491 if (res.equals("")) {
492     res = String.valueOf((char)(int) s1.pop());
493 }
494 } catch (FileNotFoundException e) {
495     e.printStackTrace();
496 } catch (IOException e) {
497     e.printStackTrace();
498 }
499 try {
500     FileWriter writer = new FileWriter(outputFile, false);
501     System.out.println("\n\nРезультат: ");
502     System.out.print(res);
503     writer.append(res);
504     writer.flush();
505 } catch (IOException e) {
506     e.printStackTrace();
507 }
508 System.out.println("");
509 }
510
511 public static void equation2(String inputFile, String outputFile) {
512     System.out.println("\n\u001B[47m\u001B[30m
        10. Вычислить значение выражения:\u001B[0m");
513     String str = "";
514     String res = "";
515     Stack s1 = new Stack(1000);
516     int c;
517     try {
518         FileReader reader = new FileReader(inputFile);
519         System.out.println("Выражение: ");

```

```

520 while ((c = reader.read()) != -1) {
521     System.out.print((char) c);
522     if (c > 47 && c < 58) {
523         str = str.concat(String.valueOf((char) c));
524     }
525     if (c == 'M' || c == 'N' || c == '(') {
526         s1.push(c);
527     }
528     if (c == ',') {
529         while (s1.peek() != null && s1.peek() != '(') {
530             str = str.concat(String.valueOf((char)(int) s1.pop()));
531         }
532         if (s1.peek() == null) {
533             res = "Ошибка в выражении";
534             break;
535         }
536     }
537     if (c == ')') {
538         while (s1.peek() != null && s1.peek() != 40) {
539             str = str.concat(String.valueOf((char)(int) s1.pop()));
540         }
541         if (s1.peek() == null) {
542             res = "Ошибка в выражении";
543             break;
544         }
545         if (s1.peek() == '(') {
546             s1.pop();
547         }
548         if (s1.peek() == 'M' || s1.peek() == 'N') {
549             str = str.concat(String.valueOf((char)(int) s1.pop()));
550         }
551     }
552     if (c == 'A' || c == 'X' || c == 'O') {
553         while (s1.peek() != null && (s1.peek() == 'N' ||
554             s1.peek() == 'A' || s1.peek() == 'X' ||
555             s1.peek() == 'O')) {
556             str = str.concat(String.valueOf((char)(int) s1.pop()));
557         }
558         s1.push(c);
559     }
560 }
561 while (s1.peek() != null) {
562     if (s1.peek() == '(') {
563         res = "Ошибка в выражении";
564         break;
565     }
566     str = str.concat(String.valueOf((char)(int) s1.pop()));
567 }
568 for (int i = 0; i < str.length(); i++) {
569     if (str.charAt(i) > 47 && str.charAt(i) < 58) {
570         s1.push((int) str.charAt(i));
571     }

```

```

570         if (str.charAt(i) == 'M') {
571             int b1, b2;
572             b1 = s1.pop();
573             b2 = s1.pop();
574             s1.push(Math.max(b1, b2));
575         }
576         if (str.charAt(i) == 'N') {
577             int b1, b2;
578             b1 = s1.pop();
579             b2 = s1.pop();
580             s1.push(Math.min(b1, b2));
581         }
582     }
583     if (res.equals("")) {
584         res = String.valueOf((char)(int) s1.pop());
585     }
586 } catch (FileNotFoundException e) {
587     e.printStackTrace();
588 } catch (IOException e) {
589     e.printStackTrace();
590 }
591 try {
592     FileWriter writer = new FileWriter(outputFile, false);
593     System.out.println("\n\nРезультат: ");
594     System.out.print(res);
595     writer.append(res);
596     writer.flush();
597 } catch (IOException e) {
598     e.printStackTrace();
599 }
600 System.out.println("");
601 }
602
603 public static void equation3(String inputFile, String outputFile) {
604     System.out.println("\n\u001B[47m\u001B[30m
        11. Проверить выражение:\u001B[0m");
605     String str = "";
606     String res = "Выражение корректно";
607     Stack s1 = new Stack(1000);
608     int c;
609     try {
610         FileReader reader = new FileReader(inputFile);
611         System.out.println("Выражение: ");
612         while ((c = reader.read()) != -1) {
613             System.out.print((char) c);
614             if (c == 'x' || c == 'y' || c == 'z') {
615                 str = str.concat(String.valueOf((char) c));
616             }
617             if (c == '(') {
618                 s1.push(c);
619             }
620             if (c == ')') {

```

```

621         while (s1.peek() != null && s1.peek() != 40) {
622             str = str.concat(String.valueOf((char)(int) s1.pop()));
623         }
624         if (s1.peek() == null) {
625             res = "Ошибка в выражении";
626             break;
627         }
628         if (s1.peek() == '(') {
629             s1.pop();
630         }
631         if (s1.peek() == 'M' || s1.peek() == 'N') {
632             str = str.concat(String.valueOf((char)(int) s1.pop()));
633         }
634     }
635     if (c == '+' || c == '-') {
636         while (s1.peek() != null && (s1.peek() == '+' ||
637             s1.peek() == '-')) {
638             str = str.concat(String.valueOf((char)(int) s1.pop()));
639         }
640         s1.push(c);
641     }
642     while (s1.peek() != null) {
643         if (s1.peek() == '(') {
644             res = "Ошибка в выражении";
645             break;
646         }
647         str = str.concat(String.valueOf((char)(int) s1.pop()));
648     }
649     for (int i = 0; i < str.length(); i++) {
650         if (str.charAt(i) > 47 && str.charAt(i) < 58) {
651             s1.push((int) str.charAt(i));
652         }
653         if (str.charAt(i) == 'M') {
654             int b1, b2;
655             b1 = s1.pop();
656             b2 = s1.pop();
657             s1.push(Math.max(b1, b2));
658         }
659         if (str.charAt(i) == 'N') {
660             int b1, b2;
661             b1 = s1.pop();
662             b2 = s1.pop();
663             s1.push(Math.min(b1, b2));
664         }
665     }
666 } catch (FileNotFoundException e) {
667     e.printStackTrace();
668 } catch (IOException e) {
669     e.printStackTrace();
670 }
671 try {

```

```
672     FileWriter writer = new FileWriter(outputFile, false);
673     System.out.println("\n\nРезультат: ");
674     System.out.print(res);
675     writer.append(res);
676     writer.flush();
677 } catch (IOException e) {
678     e.printStackTrace();
679 }
680 System.out.println("");
681 }
682 }
```

4. ВЫВОД

В результате выполнения данной лабораторной работы изучены и реализованы такие структуры данных как стек и дек. С помощью реализованных структур данных выполнены различные задания.