

## Лабораторна робота №6

Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача

**Мета:** Тривале зберігання та відновлення стану об'єктів. Ознайомлення з принципами серіалізації/десеріалізації об'єктів. Використання бібліотек класів користувача.

### 1. Вимоги:

1. Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибірккову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
4. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
5. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

#### 1.1. Розробник:

- Кедровський Максим
- КІТ-119а
- 10 варіант

#### 1.2. Загальне завдання:

Забезпечити серіалізацію/десеріалізацію контейнеру.

### 2. Опис програми:

```

1 package task06;
2
3 import java.io.FileInputStream;
4
5
6
7
8
9 public class Ser {
10     public static boolean writeContainer(Container Obj,String fileName) {
11         try {
12             FileOutputStream fos = new FileOutputStream(fileName);
13             ObjectOutputStream oos = new ObjectOutputStream(fos);
14
15             oos.writeObject(Obj);
16             oos.close();
17             return true;
18         } catch (IOException e) {
19             e.printStackTrace();
20             return false;
21         }
22     }
23     public static Container readContainer(Container Obj,String fileName) {
24         try {
25             FileInputStream fis = new FileInputStream(fileName);
26             ObjectInputStream ois = new ObjectInputStream(fis);
27             Obj = (Container)ois.readObject();
28             ois.close();
29             return Obj;
30
31         } catch (IOException e) {
32             e.printStackTrace();
33             return null;
34         } catch (ClassNotFoundException e) {
35             // TODO Auto-generated catch block
36             e.printStackTrace();
37             return null;
38         }
39     }
40 }
41

```

Рисунок 6.1 - Код Ser.java

```

1 package task06;
2
3 import java.io.FileInputStream;
4
5
6
7
8
9
10
11 public class Main {
12     public static void main(String[] args) {
13         Container cont = new Container();
14         String fileName;
15         String text;
16         Scanner in = new Scanner(System.in);
17         while(true) {
18             System.out.print("1 - Занесение информации. \n"
19                 + "2 - Сохранение контейнера.\n"
20                 + "3 - Чтение контейнера.\n"
21                 + "4 - Вывод содержимого контейнера.\n"
22                 + "5 - Добавление элемента. \n"
23                 + "6 - Удаление элемента. \n"
24                 + "7 - Очистка контейнера.\n"
25                 + "8 - Проверка наличия элемента в контейнере.\n"
26                 + "9 - Сортировка\n"
27                 + "10 - Поиск слова\n"
28                 + "11 - Выход из программы. \n");
29             System.out.print("Выберите команду: \n");
30             int number = Integer.parseInt(in.nextLine());
31             switch (number) {
32                 case 1:
33                     System.out.println("Введите текст:");
34                     text = in.nextLine();
35                     cont = new Container(text);
36                     break;
37                 case 2:
38                     System.out.println("Выберите файл для записи:");
39                     fileName = in.nextLine();
40                     Ser.writeContainer(cont, fileName);
41                     break;
42                 case 3:
43                     System.out.println("Выберите файл для чтения:");
44                     fileName = in.nextLine();
45                     cont = Ser.readContainer(cont, fileName);
46                     break;
47                 case 4:
48                     System.out.println(cont.toString());
49                     break;

```

Рисунок 6.2 - Код Main.java

```

50         case 5:
51             System.out.println("Введите текст:");
52             text = in.nextLine();
53             cont.add(text);
54             break;
55         case 6:
56             System.out.println("Введите текст:");
57             text = in.nextLine();
58             cont.remove(text);
59             break;
60         case 7:
61             cont.clear();
62             break;
63         case 8:
64             System.out.println("Введите текст:");
65             text = in.nextLine();
66             if (cont.containsAll(new Container(text))) System.out.println("Контейнер содержит указанные слова");
67             else System.out.println("Контейнер не содержит все указанные слова");
68             break;
69         case 9:
70             cont.sort();
71             break;
72         case 10:
73             System.out.println("Введите текст:");
74             text = in.nextLine();
75             if (cont.contains(text) != -1) System.out.println("Указанное слово на " + cont.contains(text) + " месте в контейнере");
76             else System.out.println("Нет такого элемента в контейнере");
77             break;
78         case 11:
79             return;
80     }
81 }
82 }
83 }

```

Рисунок 6.3 - Код Main.java 2

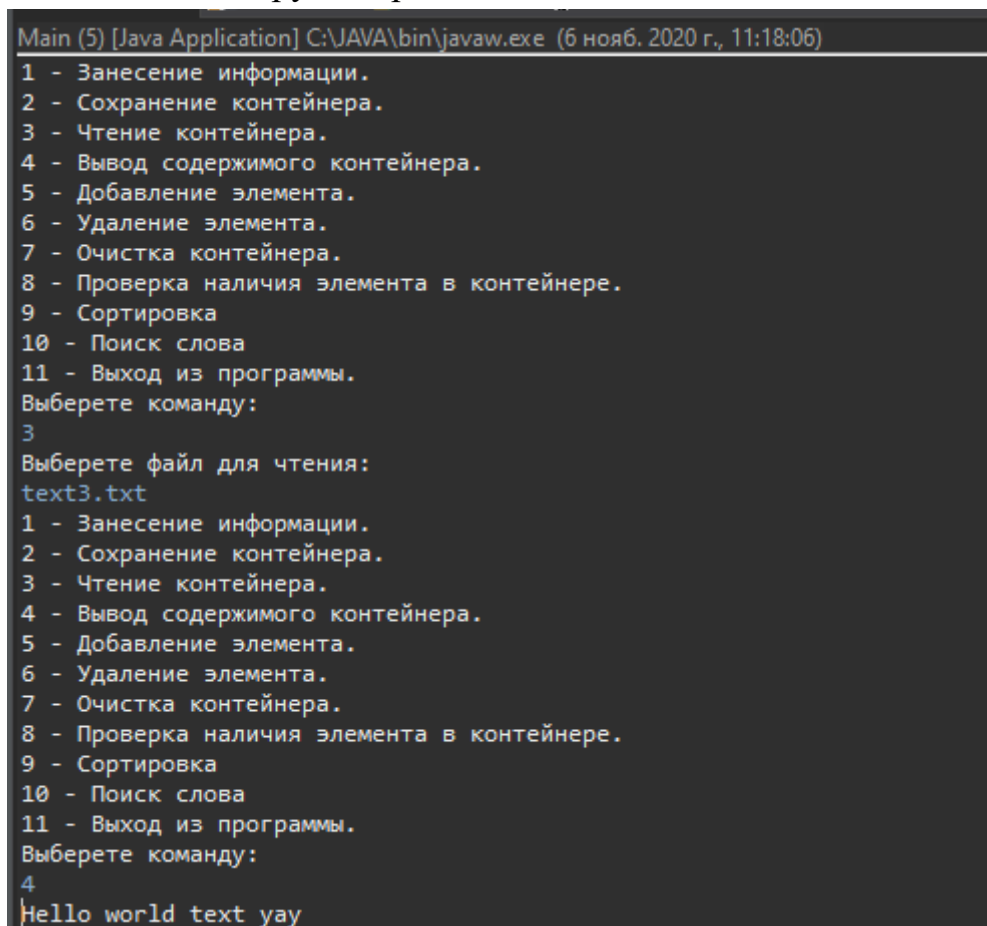
У класі Main за допомогою циклу while та switch(number) реалізовано взаємодію з користувачем шляхом вводу команд:

- 1 - Занесение информации
- 2 - Сохранение контейнера
- 3 - Чтение контейнера
- 4 - Вывод содержимого контейнера
- 5 - Добавление элемента
- 6 - Удаление элемента
- 7 - Очистка контейнера
- 8 - Проверка наличия элемента в контейнере
- 9 - Сортировка
- 10 - Поиск слова
- 11 - Выход из программы

Серіалізація реалізована у класі Ser, у методах writeContainer, readContainer, що записують та читають контейнер з файлу відповідно.

### 3. Варіанти використання:

Програма призначена для серіалізації/десеріалізації раніше розробленого контейнеру. Наприклад:



```
Main (5) [Java Application] C:\JAVA\bin\javaw.exe (6 нояб. 2020 г., 11:18:06)
1 - Занесение информации.
2 - Сохранение контейнера.
3 - Чтение контейнера.
4 - Вывод содержимого контейнера.
5 - Добавление элемента.
6 - Удаление элемента.
7 - Очистка контейнера.
8 - Проверка наличия элемента в контейнере.
9 - Сортировка
10 - Поиск слова
11 - Выход из программы.
Выберете команду:
3
Выберете файл для чтения:
text3.txt
1 - Занесение информации.
2 - Сохранение контейнера.
3 - Чтение контейнера.
4 - Вывод содержимого контейнера.
5 - Добавление элемента.
6 - Удаление элемента.
7 - Очистка контейнера.
8 - Проверка наличия элемента в контейнере.
9 - Сортировка
10 - Поиск слова
11 - Выход из программы.
Выберете команду:
4
Hello world text yay
```

Рисунок 6.4 - Демонстрація роботи з програмою

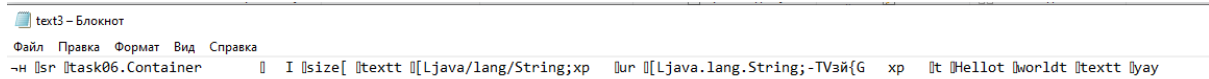


Рисунок 6.5 - Вміст файлу text3.txt

### **Висновок:**

Реалізував тривале зберігання та відновлення стану об'єктів. Ознайомився з принципами серіалізації/десеріалізації об'єктів. Використав бібліотеку класів користувача. Мовою Java було розроблено програму відповідно до індивідуального завдання.