

## **6. Сериалізація/Десериалізація об'єктів. Бібліотека класів користувача**

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

### **1 ВИМОГИ**

#### **1.1 Розробник**

Інформація про розробника:

- Кононенко Дмитро Олексійович
- НТУ “ХПІ” 1.KIT102.8a
- Варіант 08

#### **1.2 Загальне завдання**

1. Реалізувати і продемонструвати тривале зберігання/відновлення “раніше розробленого контейнера” за допомогою серіалізації/десериалізації.
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.

#### **1.3 Задача**

**Варіант 8.** Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом навпаки) іншого (наприклад: “abc”-“cba”, “def”-“fed”). Результат вивести у вигляді таблиці.

## 2 ОПИС ПРОГРАМИ

### 2.1 Засоби ООП

У даній програмі відсутні об'єктно-орієнтовані методи.

### 2.2 Ієрархія та структура даних

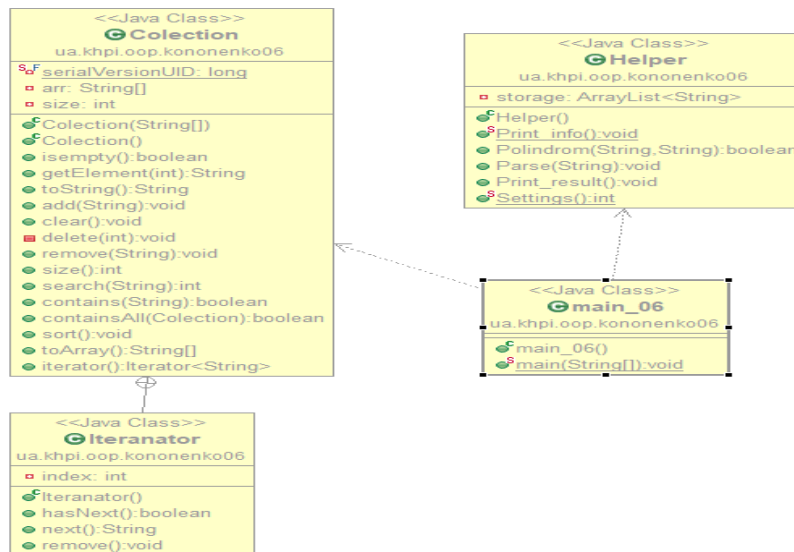


Рисунок 1 – Діаграма класів

### 2.3 Важливі фрагменти програми

```
52     case 4:
53         if(!test.isEmpty()) {
54             FileOutputStream fileOutputStream = new FileOutputStream("D:/test2.txt");
55             ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
56             objectOutputStream.writeObject(test);
57             objectOutputStream.close();
58         }
59         break;
60     case 5:
61
62         FileInputStream fileInputStream = new FileInputStream("D:/test.txt");
63         ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
64
65         test = (Collection) objectInputStream.readObject();
66         objectInputStream.close();
67
68         break;
69
```

Рисунок 2 – Реалізація стандартного протоколу серіалізації та десеріалізації

```

15 public boolean Polindrom(final String arg_1,final String arg_2) {
16
17     if(arg_1.length() != arg_2.length() | arg_1.equals("")) {
18         return false;
19     }
20
21     for (int i = 0; i < arg_1.length(); i++) {
22         char ch_1 = arg_1.charAt(i);
23         char ch_2 = arg_2.charAt(arg_1.length() - 1 - i);
24         if(ch_1 != ch_2) {
25             return false;
26         }
27     }
28     return true;
29 }
30
31
32 public void Parse(String parse_string){
33     ArrayList<String> storage2 = new ArrayList<String>();
34     String temp_storage;
35     int temp = parse_string.indexOf(" ");
36     temp++;
37     while(parse_string.length() != 0) {
38
39         temp_storage = new String(parse_string.substring(0,temp));
40         temp_storage = temp_storage.trim();
41
42         storage2.add(temp_storage);
43
44         parse_string = new String(parse_string.substring(temp));
45
46         temp = parse_string.indexOf(" ")+1;
47         if(temp == 0) { // when the program trying to find last word
48
49             temp = parse_string.length();
50
51         }
52     }
53
54
55
56     storage = storage2;
57 }

```

Рисунок 4 – метод Parse() та Polindrom()

### 3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє виконувати серіалізацію об'єктів контейнера у файл, та десеріалізовувати вихідні дані з файлу до відновлення об'єктів створеного контейнера. Також програма виконує сортування, пошук елементу у контейнері, а також обробку об'єктів (рядків), якщо вони є у контейнері.

```
0 - Exit
1 - Insert data
2 - Show data
3 - Parse data and show result
4 - Save collection
5 - Download collection
6 - Sort
7 - Search
Select:
5

0 - Exit
1 - Insert data
2 - Show data
3 - Parse data and show result
4 - Save collection
5 - Download collection
6 - Sort
7 - Search
Select:
2
Hello it is me mario and oiram
It is gonna be hot here si
We should find better place
Oh hello there olleh
```

Рисунок 3 – Результати роботи *десеріалізації()*

```
Oh hello there olleh
It is gonna be hot here si
We should find better place
Hello it is me mario and oiram

0 - Exit
1 - Insert data
2 - Show data
3 - Parse data and show result
4 - Save collection
5 - Download collection
6 - Sort
7 - Search
Select:
```

Рисунок 4 – Результати роботи методу *sortArray()*

```
0 - Exit
1 - Insert data
2 - Show data
3 - Parse data and show result
4 - Save collection
5 - Download collection
6 - Sort
7 - Search
Select:
7
Input string to search:
We should find better place
Index of string: 2
```

Рисунок 5 – Результати роботи методу *getIndex()*

## ВИСНОВКИ

В даній лабораторній роботі розробив та реалізував методи сортування, пошуку, а також обробки елементів контейнера. Також успішно виконав серіалізацію та десеріалізацію об'єктів контейнера.