

## **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**

#### **1.1 Розробник**

Інформація про розробника:

- Кулик Данііл Ігорович
- НТУ “ХП” 1.KIT102.8a
- Варіант 11

#### **1.2 Загальне завдання**

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.

#### **1.3 Задача**

11. Ввести текст. У тексті слова заданої довжини замінити зазначеним рядком. Вивести початковий текст та результат.

### **2 ОПИС ПРОГРАМИ**

#### **2.1 Засоби ООП**

У даній програмі відсутні об'єктно-орієнтовані методи.

## 2.2 Ієрархія та структура даних

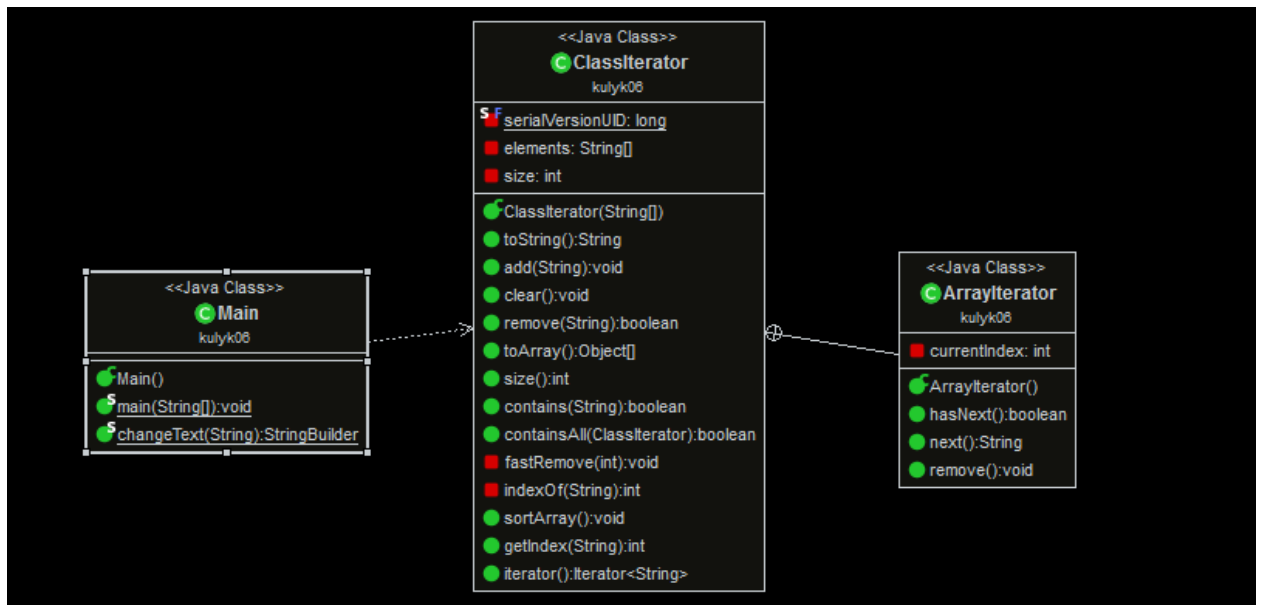


Рисунок 1 – Діаграма класів

## 2.3 Важливі фрагменти програми

```
File f = new File("C:\\Users\\Daniil\\eclipse-workspace\\kulyk-daniil\\src\\Test.txt");
FileOutputStream fileOutputStream = new FileOutputStream(f);
ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
objectOutputStream.writeObject(text);
objectOutputStream.close();

FileInputStream fileInputStream = new FileInputStream(f);
ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
ClassIterator newText = (ClassIterator) objectInputStream.readObject();
objectInputStream.close();
System.out.println(newText.toString());
```

Рисунок 2 – Реалізація стандартного протоколу серіалізації та десеріалізації

```

public static StringBuilder changeText(String string) {
    System.out.print("Enter which word replace current:");
    Scanner in = new Scanner(System.in);
    String tmp = in.next();
    StringBuilder word = new StringBuilder();
    word.append(tmp);
    StringBuilder result = new StringBuilder(string.length());
    result.append(string);

    for (int i = 0, j = 0; i < string.length(); i++) {
        if (string.charAt(i) != ' ') {
            j = i;
            while (string.charAt(j) != ' ') {
                word.append(string.charAt(j));
                j++;
            }
            if (word.length() == tmp.length()) {
                result.append(tmp + " ");
                i = j;
            } else {
                for (; i < j; i++) {
                    result.append(string.charAt(i));
                }
                result.append(" ");
            }
            word.delete(0, word.length());
        }
    }

    return result.delete(0, result.length()/2);
}

```

Рисунок 4 – метод *changeText()*

### 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.

```
Menu:
1.Show contents
2.Add new text
3.Remove all of elements
4.Remove found element
5.Show array
6.Show amount of elements in array
7.Check for a specific element
8.Check for source elements
9.Iterator 'while' cycle
10.Iterator 'for each' cycle
11.Change an element
12.Sort elements in container
13.Find required element
14.Exit program

Select option:
1
[ "Help me, please " "I really need some help " "OK, gg, boys " ]
```

Рисунок 3 – Результати роботи методу *toString()*

```
Menu:
1.Show contents
2.Add new text
3.Remove all of elements
4.Remove found element
5.Show array
6.Show amount of elements in array
7.Check for a specific element
8.Check for source elements
9.Iterator 'while' cycle
10.Iterator 'for each' cycle
11.Change an element
12.Sort elements in container
13.Find required element
14.Exit program

Select option:
12
[ "Help me, please " "I really need some help " "OK, gg, boys " ]
```

Рисунок 4 – Результати роботи методу *sortArray()*

```
Menu:
1.Show contents
2.Add new text
3.Remove all of elements
4.Remove found element
5.Show array
6.Show amount of elements in array
7.Check for a specific element
8.Check for source elements
9.Iterator 'while' cycle
10.Iterator 'for each' cycle
11.Change an element
12.Sort elements in container
13.Find required element
14.Exit program

Select option:
13
Enter the element of container:
Help me, please
The index of element is:
0
```

---

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.