

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

- Абдулаєв Ібрагім Заурбекович
- Група КІТ-119В
- Варіант 1

2 ОПИС ПРОГРАМИ

2.1 Опис класів

- **ua.khpi.oop.abdulaev06.Main**

<i>Method Detail</i>	
waitForEnter	
<pre>public static void waitForEnter()</pre>	
Очікування клавіші Enter	
clearConsole	
<pre>public static void clearConsole()</pre>	
Очищення вікна консолі	
appMenu	
<pre>public static void appMenu()</pre>	
Головне меню програми	
main	
<pre>public static void main(java.lang.String[] args)</pre>	

Рис. 2.1 – Опис класу **Main**

Текст програми

Main.java

```
package ua.khpi.oop.abdulaev06;

import ua.khpi.oop.abdulaev04.Helper;
import ua.khpi.oop.abdulaev05.Container;

import java.io.IOException;
import java.util.Scanner;

public class Main {
    public static void waitEnter() {
        try {
            System.in.read();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    /**
     * Очищення вікна консолі
     */
    public static void clearConsole() {
        System.out.print("\033[H\033[2J");
        System.out.flush();
    }

    public static void appMenu() {
        Container container = new Container();
        Scanner sc = new Scanner(System.in);
        String menuStr;
        int menuOption = 0, menuSortOption = 0;

        while (true) {
            clearConsole();
            System.out.println(
                "0. Вихід\n"
                + "1. Додати до контейнеру\n"
                + "2. Видалити з контейнеру\n"
                + "3. Зчитати контейнер з файлу\n"
                + "4. Записати контейнер у файл\n"
                + "5. Вивести контейнер\n"
                + "6. Сортувати контейнер за [1 - довжиною, 2
- алфавітом]\n"
                + "7. Пошук в контейнері\n"
            );
        }
    }
}
```

```

        + "8. Обробка елементів відкомпільованим
классом\n"
        + "9. Обробка елементів власним классом\n"
    );

    menuOption = sc.nextInt();

    switch (menuOption) {
        case 0: return;
        case 1:
            sc.nextLine();
            menuStr = sc.nextLine();
            container.add(menuStr);
            break;
        case 2:
            sc.nextLine();
            menuStr = sc.nextLine();
            container.remove(menuStr);
            break;
        case 3:
            sc.nextLine();
            menuStr = sc.nextLine();
            container.read(menuStr);
            waitForEnter();
            break;
        case 4:
            sc.nextLine();
            menuStr = sc.nextLine();
            container.write(menuStr);
            waitForEnter();
            break;
        case 5:
            System.out.println(container.toString());
            waitForEnter();
            break;
        case 6:
            sc.nextLine();
            menuSortOption = sc.nextInt();
            container.sort(menuSortOption);
            break;
        case 7:
            sc.nextLine();
            menuStr = sc.nextLine();
            System.out.println("Результати пошуку:");
            for (String s : container.find(menuStr))
                System.out.println(s);
            waitForEnter();
            break;
        case 8:
            sc.nextLine();
            System.out.println("Введіть текст:");

```

```

        String data1 = sc.nextLine();
        System.out.println("Введіть слово для
заміни:");

        String data2 = sc.nextLine();
        System.out.println("Введіть довжину слів для
заміни:");

        int data3 = sc.nextInt();

        ua.khpi.oop.zhukov04.Helper.replaceByWordLength(data1, data2,
data3);

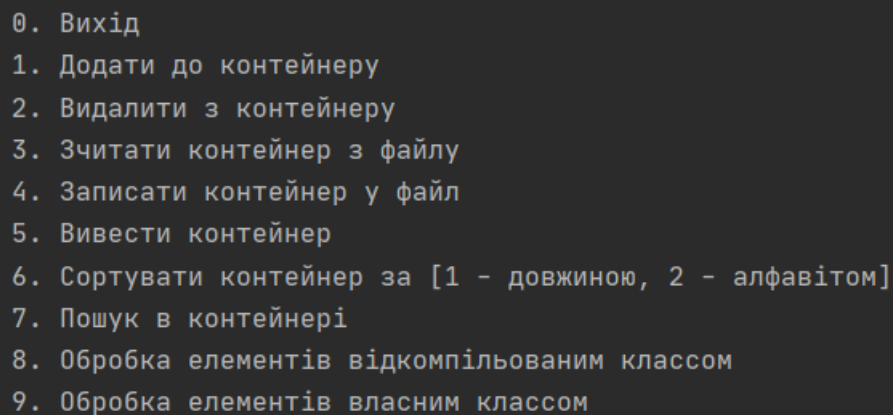
        waitForEnter();
        break;
    case 9:
        Helper.stringsInfo((String[])
container.toArray());
        break;
    }
}

}

public static void main(String[] args) {
    appMenu();
}
}

```

3 ВИСНОВКИ



```

0. Вихід
1. Додати до контейнеру
2. Видалити з контейнеру
3. Зчитати контейнер з файлу
4. Записати контейнер у файл
5. Вивести контейнер
6. Сортувати контейнер за [1 - довжиною, 2 - алфавітом]
7. Пошук в контейнері
8. Обробка елементів відкомпільованим класом
9. Обробка елементів власним класом

```

Рис. 3.1 – Головне меню програми

```
1
Ivanov Ivan
0. Вихід
1. Додати до контейнеру
2. Видалити з контейнеру
3. Зчитати контейнер з файлу
4. Записати контейнер у файл
5. Вивести контейнер
6. Сортувати контейнер за [1 - довжиною, 2 - алфавітом]
7. Пошук в контейнері
8. Обробка елементів відкомпільованим класом
9. Обробка елементів власним класом

5
Ivanov Ivan
```

Рис. 3.2 – Виведення контейнеру

```
6
|2
0. Вихід
1. Додати до контейнеру
2. Видалити з контейнеру
3. Зчитати контейнер з файлу
4. Записати контейнер у файл
5. Вивести контейнер
6. Сортувати контейнер за [1 - довжиною, 2 - алфавітом]
7. Пошук в контейнері
8. Обробка елементів відкомпільованим класом
9. Обробка елементів власним класом

5
Abdulaev Ivanov dsfjdfs
```

Рис. 3.3 – Сорткування в контейнері за алфавітом

```
0. Вихід
1. Додати до контейнеру
2. Видалити з контейнеру
3. Зчитати контейнер з файлу
4. Записати контейнер у файл
5. Вивести контейнер
6. Сортувати контейнер за [1 - довжиною, 2 - алфавітом]
7. Пошук в контейнері
8. Обробка елементів відкомпільованим класом
9. Обробка елементів власним класом

9
isVowel -> true
isConsonant -> false
isVowel -> true
isConsonant -> false
isVowel -> false
isConsonant -> true
```

Рис. 3.4 – Обробка контейнера власним класом

На лабораторній роботі навчилися використовувати тривале зберігання та відновлення стану об'єктів, ознайомилися з принципами серіалізації/десеріалізації об'єктів, навчились використовувати бібліотеки класів користувача.