**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](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task05/) за допомогою [серіалізації/десеріалізації](https://docs.oracle.com/javase/8/docs/technotes/guides/serialization/index.html).
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення [задачі л.р. №3](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/#_4) з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибіркову 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**

***Виконавець:*** Сиромятников Марк

***Група:*** КІТ-119а

**Варіант:** 19

**2 ОПИС ПРОГРАМИ**

Було використано наступні засоби:

* *import java.util.Scanner* – для вводу даних.
* System.***out***.printf() – для виводу тексту у консоль.
* .close(); - закриття потоку вводу
* import java.util.NoSuchElementException – обробка відсутності елементу.
* java.io.Serializable – бібліотека серіалізації
* java.io.FileInputStream – введеня інформації до файлу
* java.io.FileOutputStream – виведення інформації з файлу

**2.1 Ієрархія та структура класів**

Було створено 1 клас MainClas, який викликає методи виконання завдання та у якому реалізована серіалізація, клас-контейнер ContainerClass, що містить у собі дані та відповідні методи, що викликаються першим класом, класс-хелпер іншого студента, який виконує задану роботу через метод, власний клас-хелпер для виконання роботи з лаб №3.

**2.2 Важливі фрагменти програми**

case 4:

FileOutputStream fs = new FileOutputStream("Сериализация.ser");

try {

ObjectOutputStream os = new ObjectOutputStream(fs);

os.writeObject(cont);

os.close();

System.out.println("Сериализация успешна\n");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

System.out.println("Сериализация неудачна\n");

}

break;

case 5:

FileInputStream fis = new FileInputStream("Сериализация.ser");

try {

ObjectInputStream ois = new ObjectInputStream(fis);

ContainerClass cont2 = (ContainerClass) ois.readObject();

ois.close();

Iterator<String> itearator = cont2.iterator();

while (itearator.hasNext()) {

System.out.println(itearator.next());

}

System.out.println("\n");

System.out.println("Десериализация успешна\n");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

System.out.println("Десериализация неудачна\n");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

System.out.println("Десериализация неудачна\n");

}

break;

case 6:

MinMaxWords.findMinMaxWords(cont.toString());

break;

public void printAll() {

for (String string : array) {

System.out.println(string);

}

System.out.println("\n");

}

public void sorting() {

for (int a = 0; a < arrLength - 1; a++) {

for (int b = a + 1; b < array.length; b++) {

if (array[a].compareTo(array[b]) > 0) {

String temp = array[a];

array[a] = array[b];

array[b] = temp;

}

}

}

}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

Нижче наведено опис пунктів меню та прикріплено відповідні зображення задля пояснення роботи. Пункти меню, відповідно, можуть бути викликані користувачем

1. Пункт меню “Show data” Викликає метод виведення на екран вмісту контейнера (рис 1).

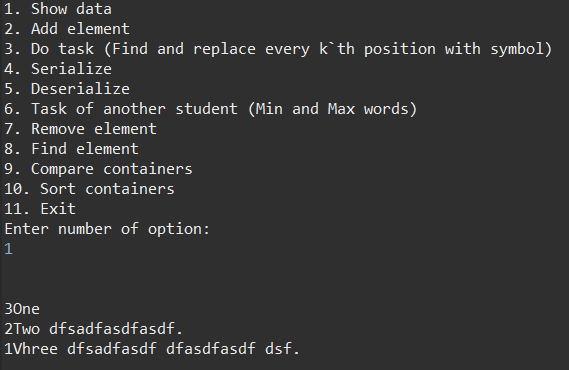


Рисунок 1 - Перший пункт меню

1. Пункт меню “Add element” викликає метод додавання нового елементу до контейнера (рис 2).

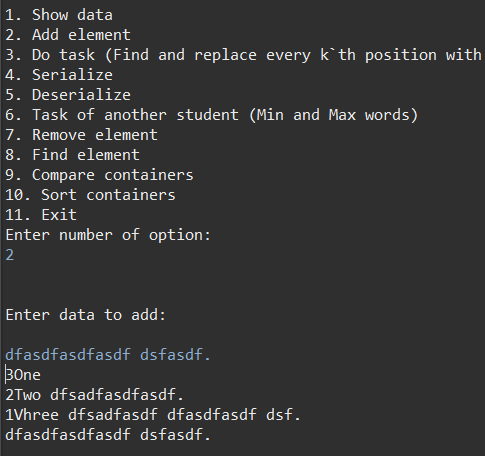


Рисунок 2 - Другий пункт меню

1. Пункт меню “Do task” викликає метод, розроблений у попередній лабораторній роботі стосовно поточних даних (рис 3).

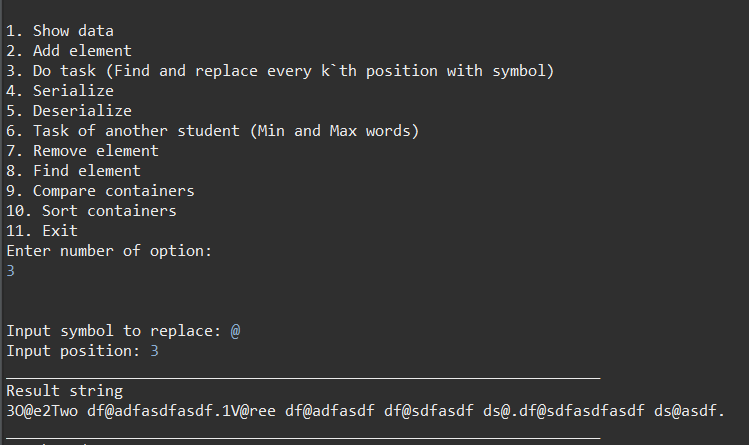


Рисунок 3 - Третій пункт меню

1. Пункт меню “Serialize” викликає метод серіалізації даних (рис 4).

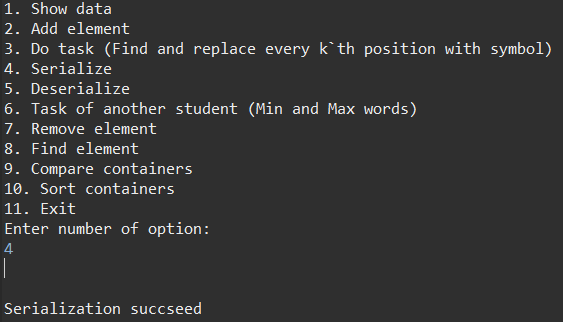


Рисунок 4 - Четвертий пункт меню

1. Пункт меню “Deserialize” викликає метод десеріалізації даних та виводу їх на екран (рис 5).

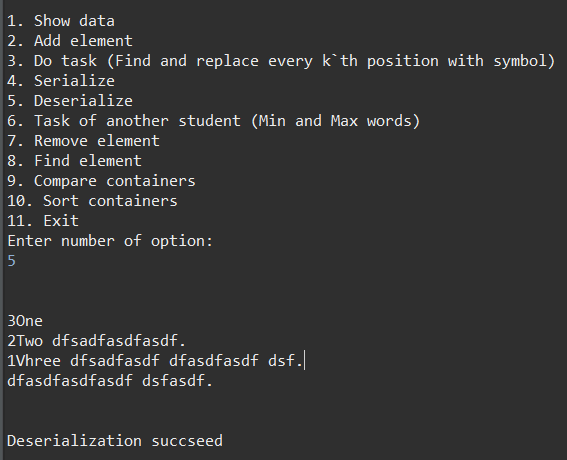


Рисунок 5 – П’ятий пункт меню

1. Пункт меню “Task of another student” викликає метод обробки даних іншого студента з jar-файлу стосовно поточних даних (рис 6).

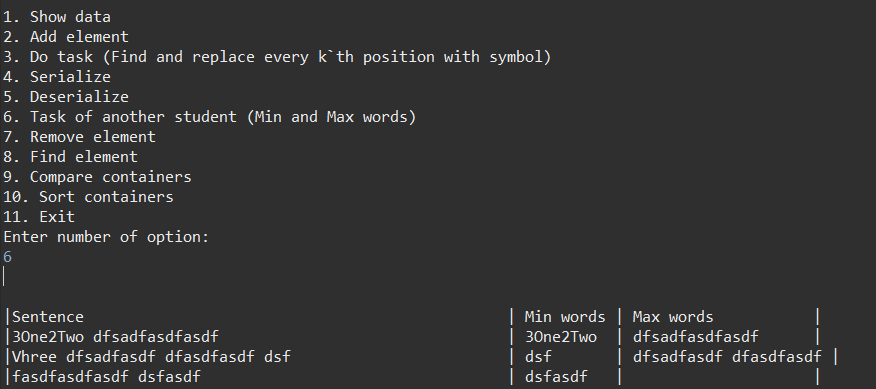


Рисунок 6 – Шостий пункт меню

1. Пункт меню “Remove element” викликає метод видалення певного елементу з контейнера за введеними даними (рис 7).

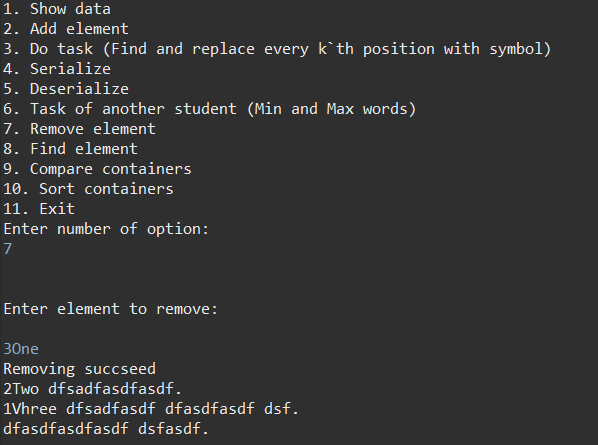


Рисунок 7 – Сьомий пункт меню

1. Пункт меню “Find element” викликає метод знаходження елемента у контейнері та повернення його номеру (рис 8).

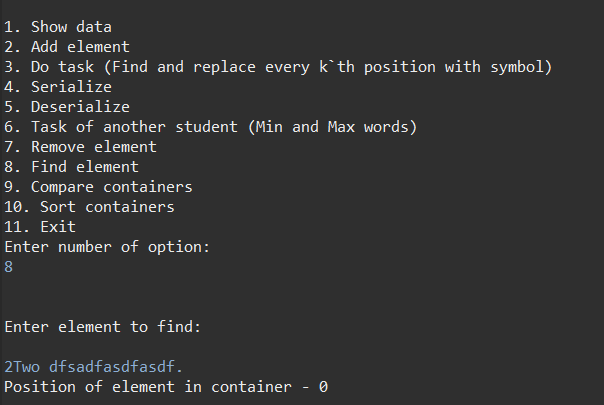


Рисунок 8 – Восьмий пункт меню

1. Пункт меню “Compare containers” викликає порівнянні даних з поточного контейнера з новим, дані якого вводить користувач (рис 9).

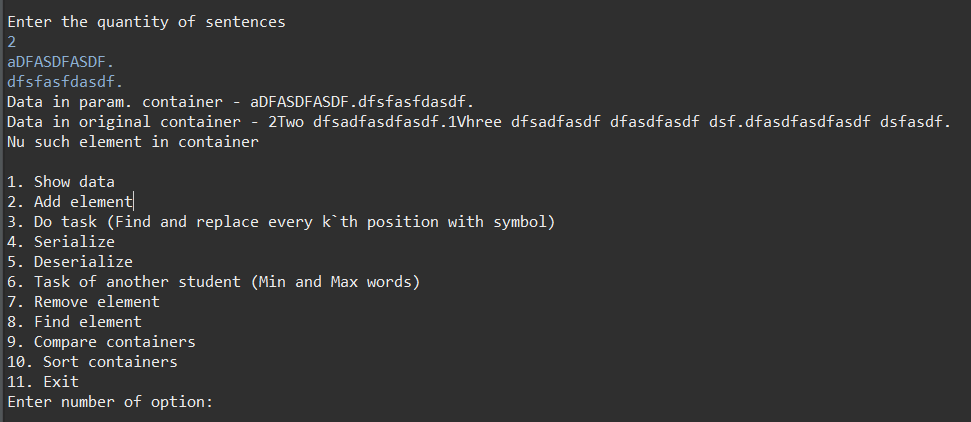


Рисунок 9 – Дев’ятий пункт меню

1. Пункт меню “Sort containers” викликає метод сортування даних контейнеру за зростанням або в алфавітному порядку (рис 10).

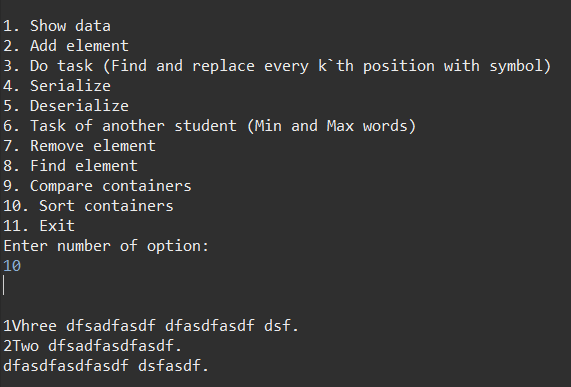


Рисунок 10 – Десятий пункт меню

1. Пункт меню “Exit” завершує роботу з програмою та перериває процес.

**ВИСНОВКИ**

У результаті виконання лабораторної роботи було розроблено методи серіалізації та десеріалізації даних з контейнеру та підключення методу іншого студента з групи. Розроблено меню для зручності з користувача. Розширено функціонал методів з попередніх лабораторних робіт.