Тема: Серіалізація/десеріалізація об’єктів. Бібліотека класів користувача.

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

**1 ВИМОГИ  
1.1 Розробник**

Інформація про розробника:

- Когутенко Олександр Олексійович;

- КІТ-119Д;

- 11 варіант.

**1.2 Загальне завдання**

* 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).
* 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/" \l "_4) з іншим студентом (визначає викладач).
* Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
* Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
* Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Використовуеться наслідування, інтерфейс, поліморфізм.

2.2 Ієрархія та структура класів

Використовую 6 классів: Array, ArrayIterator, HelperClass, InteractiveConsole, Main, SaveArray.

Array використовую як інтерфейс для класу контейнеру.

ArrayIterator використовую як особисту реалізацію ітератора.

HelperClass допоміжний класс для розрахунків.

InteractiveConsole клас для налагодженого спілкування програми з користувачем.

Main клас який має точку входу у програму.

SaveArray клас контейнер який має все необхідні методи маніпулятори.

2.3 Важливі фрагменти програми

package ua.khpi.oop.kogutenko06;  
  
  
import Laba3.Helper;  
  
import java.io.\*;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
  
*/\*\*  
 \* The type Helper class with console.  
 \*/*public class HelperClass extends Object{  
  
 .....

public void serialization(String savefile)  
{  
 File file = new File(savefile);///pathname  
 try {  
 FileOutputStream fos = new FileOutputStream(file);  
 ObjectOutputStream oos = new ObjectOutputStream(fos);  
  
 oos.writeObject(save.size());  
 System.*out*.println("size :" + save.size());  
 for (String el : save)  
 {  
 oos.writeObject(el);  
 System.*out*.println(el);  
 }  
 }  
 catch(FileNotFoundException e)  
 {  
 e.printStackTrace();  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
}

public void deserializtion(String savefile) {  
 //Array<Shop> container = new SaveArray<>();  
 File file = new File(savefile);///pathname  
 if(file.length() != 0) {  
 try {  
 FileInputStream fis = new FileInputStream(file);///pathname  
 ObjectInputStream ois = new ObjectInputStream(fis);  
 Integer count = ois.readInt();  
 for (int i = 0; i < count; i++)  
 {  
 save.add((String) ois.readObject());  
 System.*out*.println((String)ois.readObject());  
 }  
  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
 else  
 {  
 System.*out*.println("File is empty");  
 }  
 }  
  
 public void anotherClass()  
 {  
 Helper helper = new Helper();  
 StringBuilder s;  
 s = helper.*AddTextSB*();//  
 helper.*PrintLineSB*(s);//  
 s =helper.*Task6*(s);//  
 helper.*PrintNewLine*(s);//  
 }  
}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

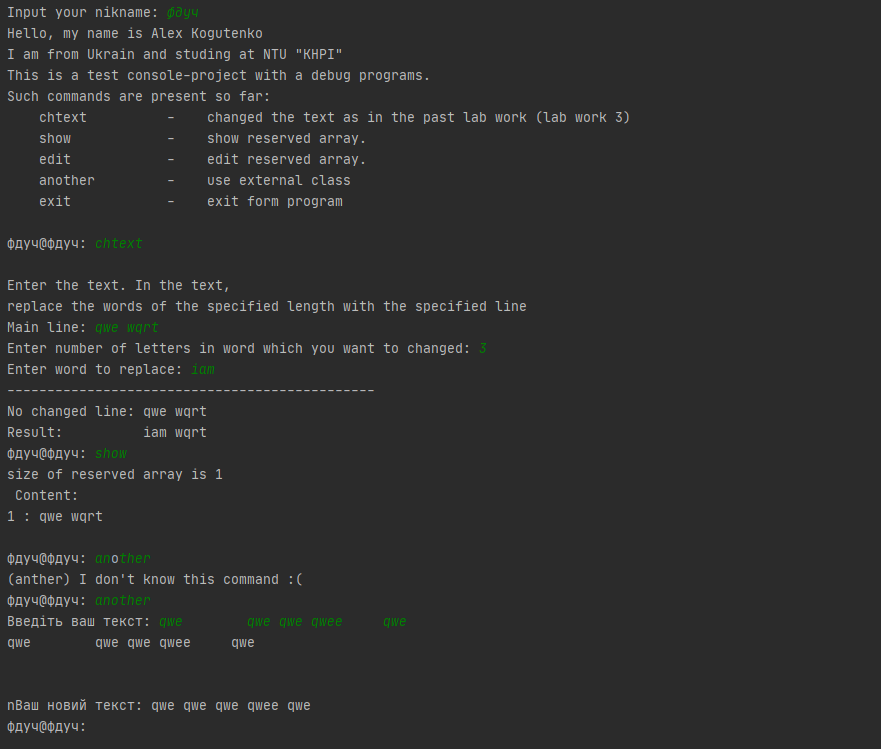


Рисунок 6.1 - робота пррограми

**ВИСНОВКИ**

Ознайомився з принципами серіалізації/десеріалізації об’єктів та використання бібліотек користувача.