**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 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. **Розробник**

* Дубина Валерій Валерійович
* Группа: КІТ-119д
* Варіант: 8

1. **ОПИС ПРОГРАМИ**
   1. **Було використано наступні засоби:**

File file = **new** File( "save.data");

FileOutputStream fos=**new** FileOutputStream(file);

– серіалізація;

File file = **new** File( "save.data");

FileInputStream fis=**new** FileInputStream(file);– десеріалізація;

* 1. **Ієрархія та структура класів**

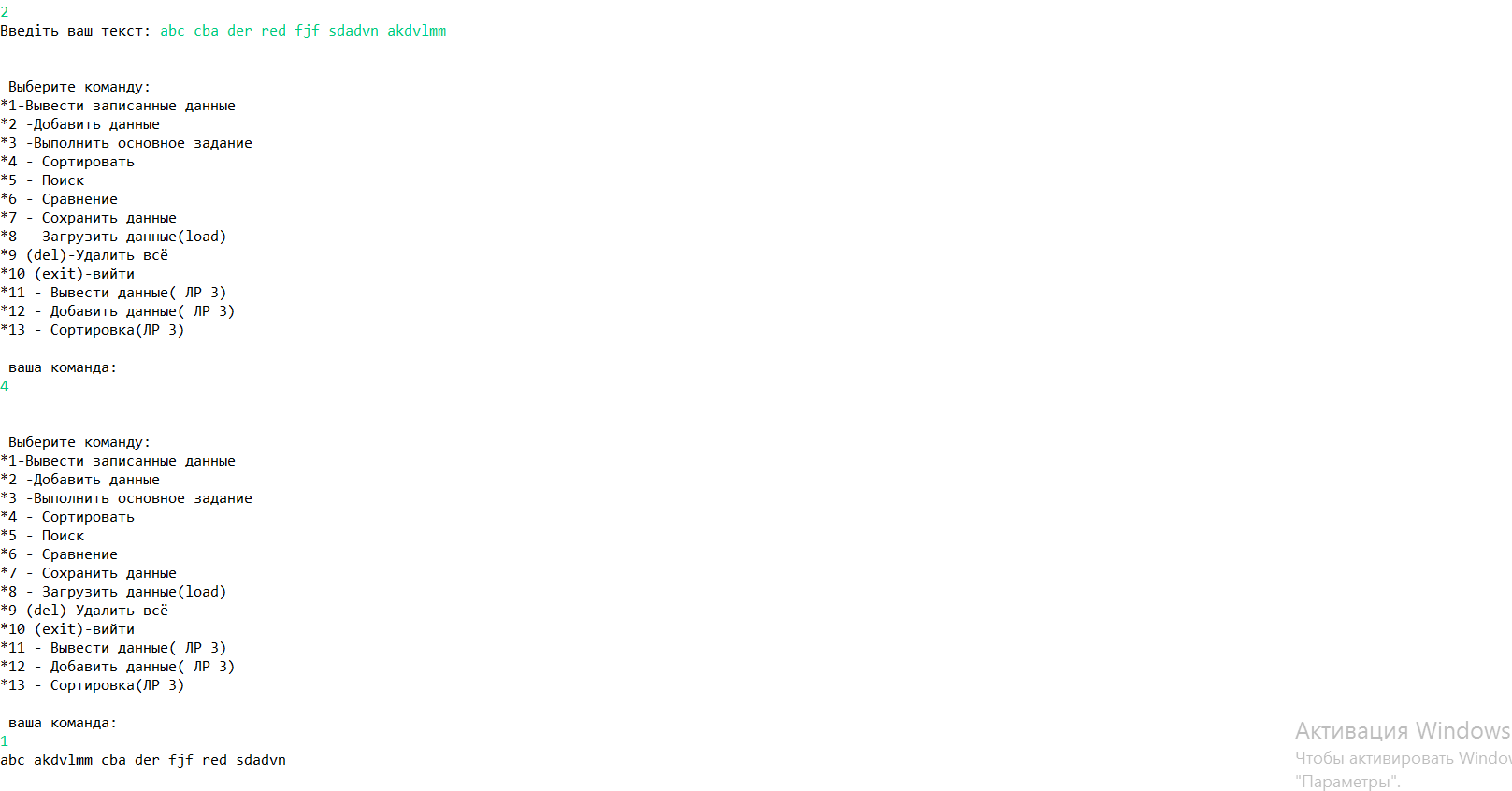
Було створено 6 класів:

* public class Main – містить метод main;
* public class Container – клас, що містить методи для роботи з контейнером;
* public class Iterator – клас, що містить методи для роботи з ітератором.
* public class Console\_program –клас,що містить меню
* public class Serializator –клас, що містить методи для серіалізації та для десеріалізації
* public class UtilityClass- клас, що виконує основне завдання
  1. **Важливі фрагменти програми**

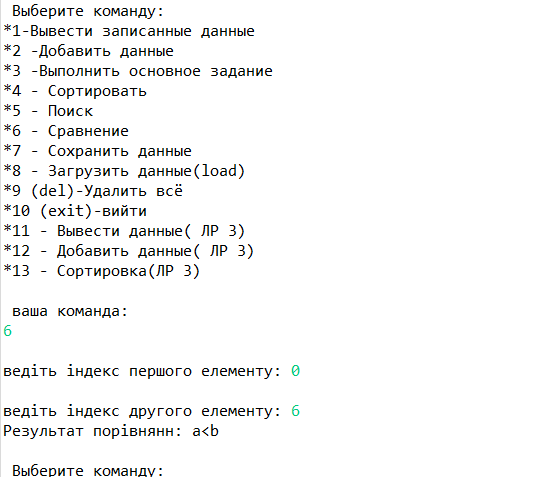
1. **public** **class** Serializator {
2. **public** **boolean** serialization(Сontainer container )
3. {
4. **boolean** flag=**false**;

7. File file = **new** File( "save.data");
8. ObjectOutputStream oos=**null**;
10. **try** {
11. FileOutputStream fos=**new** FileOutputStream(file);
12. **if**(fos!=**null**) {
13. oos= **new** ObjectOutputStream(fos);
14. oos.writeObject(container);
15. flag=**true**;
16. }
17. } **catch**(FileNotFoundException e) {e.printStackTrace();}
18. **catch** (IOException e) {e.printStackTrace(); }
19. **finally** {
20. **if**(oos!=**null**)
21. {**try** {oos.close();} **catch** (IOException e) {e.printStackTrace();}}
22. }
23. **return** flag;
24. }
25. **public** Сontainer deserializtion() {
26. Сontainer container=**null**;
27. File file = **new** File( "save.data");
28. ObjectInputStream ois=**null**;
29. **try** {
30. FileInputStream fis=**new** FileInputStream(file);
32. **if**(fis!=**null**) {
33. ois= **new** ObjectInputStream(fis);
34. container=(Сontainer)ois.readObject();
36. }
37. } **catch**(FileNotFoundException e) {e.printStackTrace();}
38. **catch** (IOException e) {e.printStackTrace(); }
39. **catch** (ClassNotFoundException e) {e.printStackTrace(); }
40. **finally** {
41. {**try** {ois.close();} **catch** (IOException e) {e.printStackTrace();}}
42. }
43. **return** container;
44. }
45. }

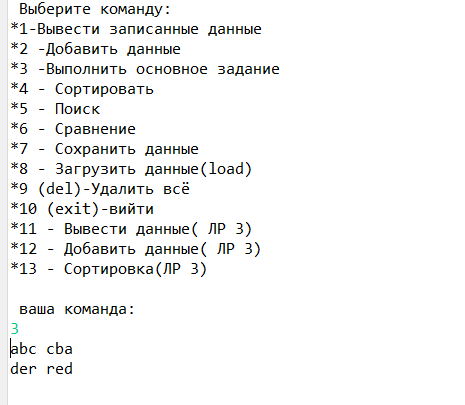
**Результати роботи програми**



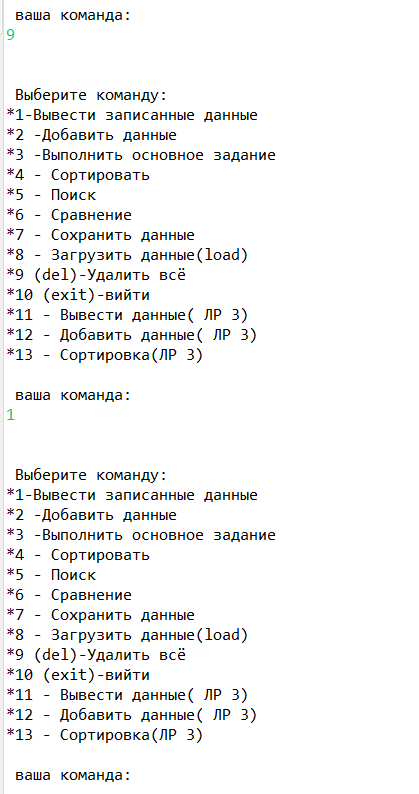
а)



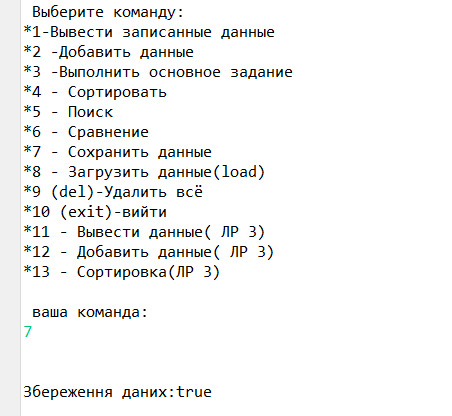
б)



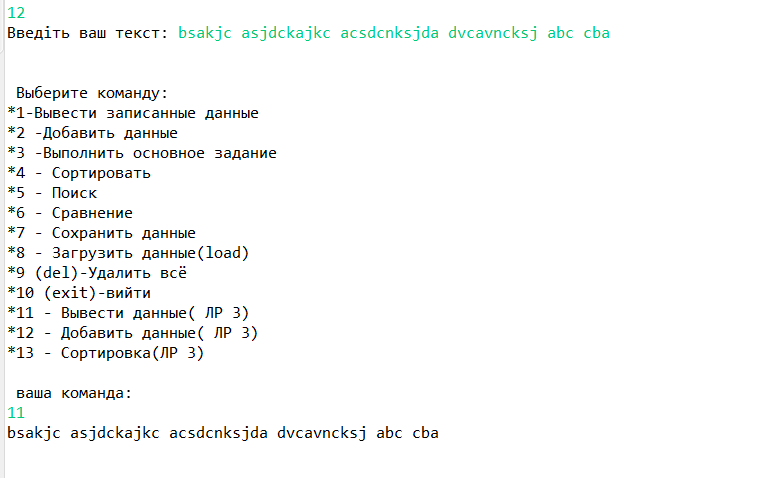
в)

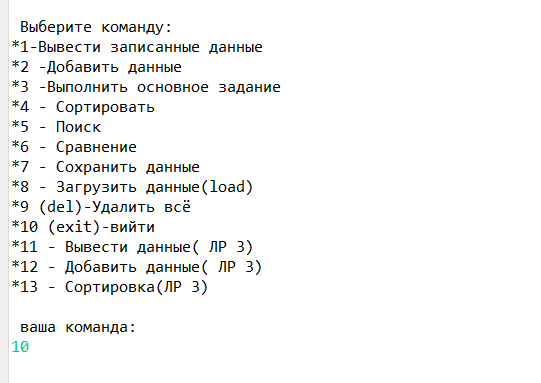


г)



д)



 е)

ж)

**Висновок**

Під час виконання лабораторної роботи було набуто навичок роботи з серіалізацією\десеріалізацієюм та з розробки бібліотеки класів користувача у середовищі Eclipse IDE.