

Звіт

Лабораторна работа 6.

Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача

Мета роботи:

- 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: Татаренко Андрій Геннадійович, КІТ-119а, варіант №20.

2. ОПИС ПРОГРАМИ

2.1. Засоби ООП: класи, методи класу, бібліотеки користувача.

2.2. Ієрархія та структура класів: публічний клас Main, клас-контейнер, та публічний клас для сереалізації.

2.3. Важливі фрагменти програми:

```
public class Serialization {

public boolean serialization(Container cont)
{

    boolean flag = false;

    File file = new File("D:/Cont.txt");

    ObjectOutputStream oos = null;

    try
    {
        FileOutputStream fos = new FileOutputStream(file);
        if(fos != null)
        {
            oos = new ObjectOutputStream(fos);
            oos.writeObject(cont);
            flag = true;
        }
    }
    catch (FileNotFoundException e)
    {
        e.printStackTrace();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    finally
    {
        if(oos != null)
        {
            try
            {
                oos.close();
            }
            catch (IOException e)
            {

```

```

        e.printStackTrace();
    }
}

return flag;
}

public Container deserialization()
{
    File file = new File("D:/Cont.txt");

    ObjectInputStream ois = null;

    Container cont = null;

    try
    {
        FileInputStream fis = new FileInputStream(file);
        if(fis != null)
        {
            ois = new ObjectInputStream(fis);
            cont = (Container)ois.readObject();
        }
    }
    catch (FileNotFoundException e)
    {
        e.printStackTrace();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    catch (ClassNotFoundException e)
    {
        e.printStackTrace();
    }
    finally
    {
        try
        {
            ois.close();
        }
        catch (IOException e)
        {

```

```

        e.printStackTrace();
    }
}

return cont;
}

}

```

2.4. Опис режимів та пунктів програми:

Пункт меню «Serialize data»: запис тексту.

Пункт меню «Deserialize data»: виконання завдання.

Пункт меню «Show container»: вивід введеного тексту.

Пункт меню «Add new element»: вивід результату завдання.

Пункт меню «Remove element»: вихід з програми.

Пункт меню «Find element»: запис тексту.

Пункт меню «Compare»: виконання завдання.

Пункт меню «Helpers task»: вивід введеного тексту.

Пункт меню «My task»: вивід результату завдання.

Пункт меню «Sort»: вихід з програми.

Пункт меню «Exit»: вихід з програми.

Результат виконання програми

```

9) My task;
10) Sort;
11) Exit;
-----
Your choice: 9
h o w      h o w
8  15 23   8  15 23
-----
1) Serialize data;
2) Deserialize data;
3) Show container;
4) Add new element;
5) Remove element;
6) Find element;
7) Compare;
8) Helpers task;
9) My task;
10) Sort;
11) Exit;
-----
Your choice: 11
Bye!!!

```

Висновки

При виконанні даної лабораторної роботи було набуто навички розробки методів для сереалізації та десереалізації. Також навчилися використовувати бібліотеки користувача.

Програма протестована, виконується без помилок.