**Звіт**

# Лабораторна работа 6.

# [Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача](https://oop-khpi.gitlab.io/#tasks)

**Мета роботи**:

* Тривале зберігання та відновлення стану об'єктів.
* Ознайомлення з принципами серіалізації/десеріалізації об'єктів.
* Використання бібліотек класів користувача.

1. **ВИМОГИ**
2. 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).
3. 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).
4. Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
5. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
6. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.
   1. **Розробник**: Татаренко Андрій Геннадійович, КІТ-119а, варіант №20.
7. **ОПИС ПРОГРАМИ**
   1. **Засоби ООП**: класи, методи класу, бібліотеки користувача.
   2. **Ієрархія та структура класів:** публічний клас Main, клас-контейнер, та публічний клас для сереалізації.
   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;

}

}

* 1. **Опис режимів та пунктів програми:**

Пункт меню «Serializate data»: запис тексту.

Пункт меню «Deserializate 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.

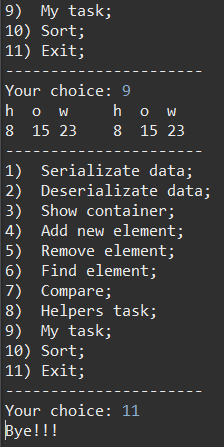
Пункт меню «Helpers task»: вивід введеного тексту.

Пункт меню «My task»: вивід результату завдання.

Пункт меню «Sort»: вихід з програми.

Пункт меню «Exit»: вихід з програми.

**Результат виконання програми**



**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.