

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

1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ 102.8(а);
- 19-12-2019р..

1.2 Загальне завдання

Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації / десеріалізації.

Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 з іншим студентом (визначає викладач).

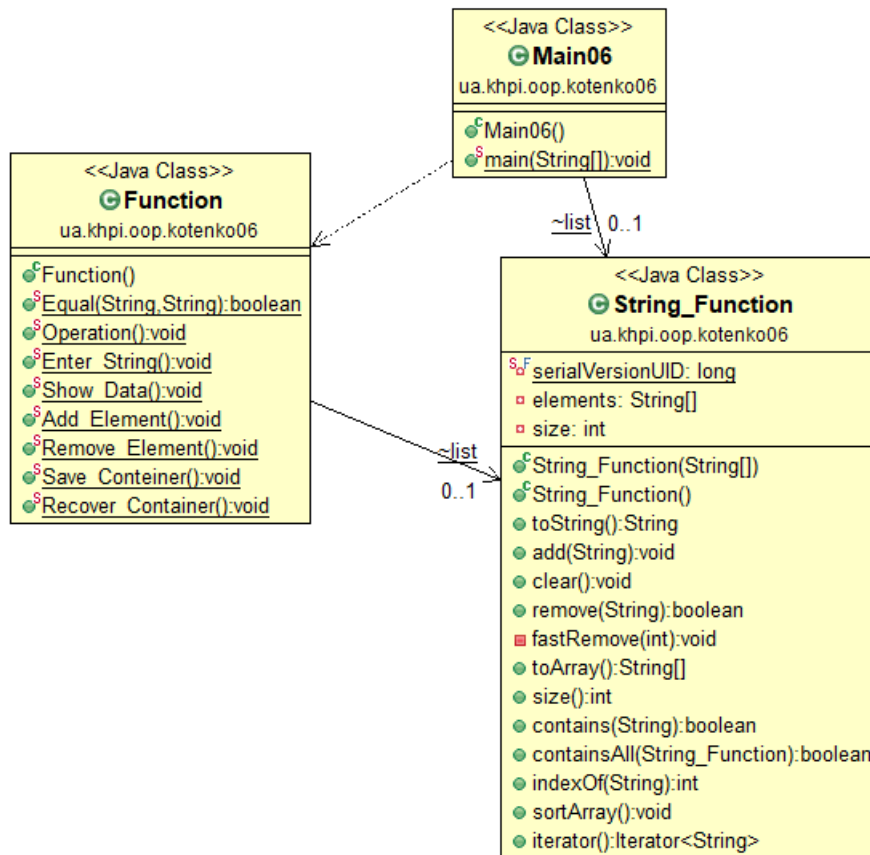
Продемонструвати послідовну та вибірккову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.

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 Опис логічної структури



2.2 Фрагменти коду

```
System.out.println("Enter your option: " + '\n'
+ "0 - Exit" + '\n'
+ "1 - Enter string" + '\n'
+ "2 - Show data" + '\n'
+ "3 - Do operation" + '\n'
+ "4 - Sort" + '\n'
+ "5 - Add element" + '\n'
+ "6 - Remove element by index" + '\n'
+ "7 - Clear" + '\n'
+ "8 - Save data" + '\n'
+ "9 - Recover data" + '\n');
```

Рисунок 2.1 – Створене діалогове меню

```

public static void Enter_String() {
    @SuppressWarnings("resource")
    Scanner in = new Scanner(System.in);
    System.out.println("Enter your string:");
    String tmp = in.nextLine();
    list.add(new String(tmp));
}

public static void Show_Data() {
    for(String tmp : list) {
        System.out.println(tmp);
    }
}

```

Рисунок 2.2 – Введення строки та вивід

```

public static void Add_Element() {
    @SuppressWarnings("resource")
    Scanner in = new Scanner(System.in);
    System.out.println("Enter your string:");
    String tmp = in.nextLine();
    list.add(tmp);
}

public static void Remove_Element() {
    @SuppressWarnings("resource")
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the text:");
    String tmp = in.nextLine();
    if(!list.contains(tmp)) {
        System.out.println("String not found");
    }
}

```

Рисунок 2.3 – Додавання та видалення елементу

```

public static void Save_Container() throws IOException {
    FileOutputStream outputStream = new FileOutputStream("C:\\eclipse\\tmp\\test.txt");
    ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);
    objectOutputStream.writeObject(list);
    objectOutputStream.close();
}

public static void Recover_Container() throws IOException, ClassNotFoundException {
    FileInputStream fileInputStream = new FileInputStream("C:\\eclipse\\tmp\\test.txt");
    @SuppressWarnings("resource")
    ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
    list = (String_Function) objectInputStream.readObject();
}

```

Рисунок 2.4 – Виведення та зчитування даних

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Ілюстрація роботи програми

```
Enter your option:  
0 - Exit  
1 - Enter string  
2 - Show data  
3 - Do operation  
4 - Sort  
5 - Add element  
6 - Remove element by index  
7 - Clear  
8 - Save data  
9 - Recover data
```

Рисунок 3.1 – Меню програми

```
2  
Hello world  
Hello World
```

Рисунок 3.2 – Виведення даних

```
-н Ёsr %ua.khpi.oop.kotenko06.String_Function  
  I Ёsize[ Ёelementst Ё[Ljava/lang/String;xp  
ur Ё[Ljava.lang.String;-TVзй{G   xp  
t ЁHello worldt ЁHello World
```

Рисунок 3.3 – Збереженні дані

ВИСНОВОК

В інтегрованому середовищі *Eclipse* розроблена програма мовою Java. Виконання програми дозволяє продемонструвати роботу зберігання та відновлення стану об'єктів, використання бібліотек класів користувача.