

## **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 (UtilityClass) 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.

**1.2 Розробник:** Завадський Дмитро Богданович КІТ119а №5.

## 1.3 Задача

Ввести текст. У тексті кожну літеру замінити її номером в алфавіті. Вивести результат наступним чином: в одному рядку друкувати текст з двома пропусками між буквами, в наступному рядку під кожною буквою друкувати її номер.

## 2 ОПИС ПРОГРАМИ

**2.1** Було використано наступні засоби:

- `Iterator<String> iterator = MyContainer.getIterator()` – Ітератор;
- `Scanner sc = new Scanner(System.in)` – для введення обраних опцій користувачем з клавіатури;
- `FileOutputStream fs = new  
FileOutputStream("Serial.ser"); ObjectOutputStream os = new  
ObjectOutputStream(fs); os.writeObject(mc); os.close()` – серіалізація;
- `FileInputStream fis = new  
FileInputStream("Serial.ser"); ObjectInputStream ois = new  
ObjectInputStream(fis); MyContainer var = (MyContainer)  
ois.readObject(); ois.close()` – десеріалізація;

## 2.2 Ієрархія та структура класів

Було створено 3 класи та використано клас розроблений у л.р. №3:

- `Public class MyContainer` – клас, що реалізує методи контейнеру.
- `Private class MyIterator` – клас, що реалізує методи ітератора.
- `public class Main` – містить лише метод `main`.

## Важливі фрагменти програми

```
public class Container implements Serializable {
    private String [] container;
    private int size;

    public String[] getContainer() {
        return container;
    }

    public void setContainer(String[] container) {
        this.container = container;
    }

    public String toString() // повертає вміст контейнера у вигляді рядка;
    {
        String str = "";
        for (String string : container) {
            str += string + " ";
        }
        return str;
    }

    public void add(String str) //додає вказаний елемент до кінця контейнеру;
    {
        int size = container.length;
        String [] new_container = new String[size+1];
        for (int i=0;i<size;i++) {
            new_container[i]=container[i];
        }
        new_container[size]=str;
        size++;
        container = new_container;
    }

    public void clear() //видаляє всі елементи з контейнеру;
    {
        for (int i = 0; i < container.length; i++) {
            container[i]=null;
        }
        size =0;
    }

    public boolean remove(String str) // видаляє перший випадок вказаного елемента
з контейнера;
    {
        boolean flag = false;
        String [] new_container = new String[size-1];
        for(int i=0;i<size;i++) {
            if(container[i].equals(str))
                flag = true;
        }
        if(flag) {
            for(int i=0,j=0;i<size;i++) {
                if(container[i].equals(str))
                    i++;
                new_container[j]=container[i];
                j++;
            }
        }
    }
}
```

```

        }
        size--;
        container = new_container;
        return flag;
    }
    else
    {
        return flag;
    }
}

public String[] toArray() //повертає масив, що містить всі елементи у
контейнері;
{
    return container;
}

public int size() //повертає кількість елементів у контейнері;
{
    return size;
}

public boolean containsAll(Container cont) //повертає true, якщо контейнер
містить всі елементи з зазначеного у параметрах;
{
    int count = 0;
    for (int i = 0; i < container.length; i++) {
        for (int j = 0; j < cont.container.length; j++) {
            if(cont.container[j].equals(container[i]))
                count++;
        }
    }
    if(count == cont.container.length)
        return true;
    else
        return false;
}

public boolean contains(String str) //повертає true, якщо контейнер містить
вказаний елемент;
{
    boolean flag = false;
    for (String string : container) {
        if(string.equals(str))
            flag=true;
    }
    return flag;
}

public void Sort() {

    String temp;

    for(int a = 0; a < size - 1; a++) {
        for(int b = a + 1; b < container.length; b++)
        {
            if(container[a].compareTo(container[b]) > 0)
            {
                temp = container[a];
                container[a] = container[b];
                container[b] = temp;
            }
        }
    }
}

```

```

    }
    }
}

public Container(String... str) {
    if(str.length!=0) {
        size = str.length;
        container = new String[size];
        for (int i=0;i<size;i++) {
            container[i]=str[i];
        }
    }
}

}

}

```

### 3 ВИСНОВКИ

Результат роботи програми:

Меню

```

Доступні команди
1 - заповнення контейнера даними -
2 - виведення змісту контейнера -
3 - додавання нового елементу в контейнер -
4 - видалення елементу з контейнера -
5 - пошук елементу -
6 - сортування контейнеру(індивідуальне завдання) -
7 - сортування контейнеру -
8 - порівняння контейнерів -
9 - серіалізація -
10 - десіалізація -
11 - завдання іншого студента -
0 - завершення програми(видалення контейнера автоматичне) -

Введіть команду:

```

## Добавление элементов

```
Введіть команду:  
1  
Введіть кількість елементів у контейнері  
3  
Введіть елементи -  
1 елемент - Move and move  
2 елемент - Slow or fast  
3 елемент - so I apologize
```

## Виконання завдань

```
Введіть команду:  
6  
Move and move  
M o v e a n d m o v e  
  
13 15 22 5 1 14 4 13 15 22 5  
  
Slow or fast  
S l o w o r f a s t  
  
19 12 15 23 15 18 6 1 19 20  
  
so I apologize  
s o I a p o l o g i z e  
  
19 15 9 1 16 15 12 15 7 9 26 5
```

Виведення змісту контейнера на екран  
Move and move Slow or fast so I apologize

```
Введіть команду:  
9  
Serialization successful
```

```
Введіть команду:  
11  
||Sentence|Vowels|Consonant|Result|  
|move and move slow or fast so i apologize|15|18|consonant|
```

## **ВИСНОВКИ**

У результаті виконання лабораторної роботи було набуто навичок роботи з серіалізацією\десеріалізацією. з розробки бібліотеки класів користувача у середовищі JavaEclipse.