

## **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:** Мітін Микита Валерійович КІТ119д №15.

### 1.3 Задача

Ввести текст. У тексті знайти та вивести всі слова-паліндроми (однаково читається в обох напрямках - зліва направо та справа наліво. Наприклад: "noon", "civic", "radar", "level", "rotor", "refer").

## 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 toString() //возвращает содержимое контейнера в виде строки;
{

    String str = "";

    for (String string : container) {

        str += string + " ";

    }

    return str;

}

public void add(String str) //добавление элемента в конец контейнера
{

    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() //удаляет все элементы контейнера
```

```

{

    for (int i = 0; i < container.length; i++) {

        container[i]=null;

    }

    size =0;

}

public boolean remove(String str) // удаление элемента
{

    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() //возвращает массив, содержащий все элементы  
    контейнера
```

```
    {  
        return container;  
    }
```

```
    public int size() //возвращает кол-во элементов
```

```
    {  
        return size;  
    }
```

```
    public boolean containsAll(Container cont) //возвращает true, если контейнер  
    содержит все элементы из указанного в 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) //возвращает true, если контейнер содержит
    указанный элемент;
    {
        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];
        }
    }
}

```

```

public Iterator<String> getIterator() {
    return new My_iterator<String>();
}

```

```

public class My_iterator<String> implements Iterator {
    int index;

    @Override
    public boolean hasNext() {
        return index < size ? true : false;
    }

    @Override
    public Object next() {
        return container[index++];
    }
}

```

/\*Method that removes from the underlying collection the last element returned by this iterator\*/

```

@Override
public void remove() {
    Container.this.remove(container[--index]);
}

```

```
}  
  
}  
  
}
```

### 3 ВИСНОВКИ

Результат роботи програми:

Меню

---

доступные команды

- 1 - заполнение контейнера данными -
- 2 - вывод содержания контейнера -
- 3 - добавление нового элемента в контейнер -
- 4 - удаление элемента с контейнера -
- 5 - поиск элемента -
- 6 - поиск полиндромов в контейнере -
- 7 - сортировка контейнера -
- 8 - сравнение контейнеров -
- 9 - сериализация -
- 10 - десериализация -
- 11 - задание другого студента -
- 0 - завершение программы(удаление контейнера автоматическое) -

Введите команду:



## Додавання елементів

---

Введіть команду:

1

введіть кількість елементів в контейнері

2

введіть елементи -

1 елемент - `as you wish rever`

2 елемент - `best radar`

доступні команди

- |      |   |   |
|------|---|---|
| 1 -  | заповнення контейнера даними                          | - |
| 2 -  | вивод вмісту контейнера                               | - |
| 3 -  | додавання нового елемента в контейнер                 | - |
| 4 -  | видалення елемента з контейнера                       | - |
| 5 -  | пошук елемента  | - |
| 6 -  | пошук поліномів в контейнері                          | - |
| 7 -  | сортування контейнера                                 | - |
| 8 -  | порівняння контейнерів                                | - |
| 9 -  | серіалізація  | - |
| 10 - | десеріалізація  | - |
| 11 - | визначення іншого студента                            | - |
| 0 -  | закінчення програми(видалення контейнера автоматичне) | - |

Введіть команду:

6

rever

radar

Виконання завдань:

Сортування та виведення

Введите команду:

7

доступные команды

- 1 - заполнение контейнера данными -
- 2 - вывод содержания контейнера -
- 3 - добавление нового элемента в контейнер -
- 4 - удаление элемента с контейнера -
- 5 - поиск элемента -
- 6 - поиск полиндромов в контейнере -
- 7 - сортировка контейнера -
- 8 - сравнение контейнеров -
- 9 - сериализация -
- 10 - десериализация -
- 11 - задание другого студента -
- 0 - завершение программы(удаление контейнера автоматическое) -

Введите команду:

2

Вывод содержимого на экран

radar rever

## Додавання рядка

Введите команду:

3

Введите новую строку

соос

Введите команду:

9

Serialization successful

## ВИСНОВКИ

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