

## **Звіт**

### **Лабораторна робота 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.

#### **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:** Момот Роман Євгенійович, КІТ119-а, варіант №14.

#### **2. ОПИС ПРОГРАМИ**

**2.1. Засоби ООП:** клас, метод класу, поле класу.

**2.2. Ієрархія та структура класів:** один публічний клас Main та публічний клас MyContainer, у полі якого знаходиться приватний клас MyIterator та публічний UtilityClass.

**2.3. Важливі фрагменти програми:**

```
public class MyContainer implements Serializable {  
    private String[] array;  
    private int size;
```

```
public int size()
{
    return size;
}

public String getLine(int index)
{
    return array[index];
}

public String toString()
{
    StringBuilder string = new StringBuilder();

    for(int i = 0; i < size; i++)
        string.append(array[i] + " ");

    return string.toString();
}

public void add(String string)
{
    String[] newArray = new String[size+1];

    for (int i = 0; i < size; i++)
        newArray[i] = array[i];

    size++;
    newArray[size - 1] = string;
    array = newArray;
}

public void clear()
```

```

{
    for(int i = 0; i < size; i++)
        array[i] = null;

    size = 0;

}

public boolean remove(String string)
{
    boolean result = false;
    int position = 0;

    for (int i = 0; i < size; i++)
        if(array[i].equals(string))
        {
            result = true;
            position = i;
            break;
        }

    if(result)
    {
        String[] newArray = new String[size-1];

        for (int i = 0; i < position; i++)
            newArray[i] = array[i];
        for (int i = position; i+1 < size; i++)
            newArray[i]=array[i+1];
    }
}

```

```

        size--;

        array=newArray;
    }

    return result;
}

public Object[] toArray()
{
    Object[] object = new Object[size];

    for (int i = 0; i < size; i++)
        object[i]=array[i];

    return object;
}

public boolean contains(String string)
{
    for (int i = 0; i < size; i++)
        if (array[i].equals(string))
            return true;

    return false;
}

public boolean containsAll(MyContainer container)
{
    boolean result = false;

    for (int i = 0; i < size; i++)
    {

```

```

        result = false;

        for (int j = 0; j < container.size(); j++)
        {
            if(array[i].equals(container.getLine(j)))
            {
                result = true;
                break;
            }
        }

        if(!result)
        {
            return false;
        }
    }

    return result;
}

public MyContainer(String... strings)
{
    if(strings.length > 0)
    {
        size = strings.length;
        array = new String[size];

        for (int i = 0; i < size; i++)
            array[i]=strings[i];
    }
}

```

```

        }
    }
    public void sort() {
        String temp;

        for(int i = 0; i < size - 1; i++)
        {
            for(int j = i + 1; j < array.length; j++)
            {
                if(array[i].compareTo(array[j]) > 0)
                {
                    temp = array[i];
                    array[i] = array[j];
                    array[j] = temp;
                }
            }
        }
    }
}

```

```

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

```

```

private class MyIterator<String> implements Iterator {
    int index;

    @Override public boolean hasNext()
    {

```

```

        if(index < size)
            return true;
        else
            return false;
    }

    @Override public Object next()
    {
        if(index==size)
            throw new NoSuchElementException();

        return array[index++];
    }

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

```

```

public void mainTask(MyContainer container)
{
    char endOfWord = '$';
    StringBuilder insertedText = new StringBuilder();
    Scanner scan = new Scanner(System.in);
    StringBuilder temp = new StringBuilder();
    int lengthOfInsertedText;
    int index = 0;

```

```
boolean stop = false;
```

```
System.out.print("\nEnter inserted text: ");
```

```
insertedText = new StringBuilder(scan.nextLine());
```

```
lengthOfInsertedText = insertedText.length();
```

```
System.out.print("Enter end of word: ");
```

```
temp = new StringBuilder(scan.next());
```

```
endOfWord = temp.charAt(0);
```

```
for(int i = 0; i < container.size(); i++)
```

```
{
```

```
    stop = false;
```

```
    temp = new StringBuilder(container.getLine(i));
```

```
    while(!stop)
```

```
    {
```

```
        index = temp.indexOf(String.valueOf(endOfWord),
```

```
index+1);
```

```
        if(index == -1)
```

```
            stop = true;
```

```
        else
```

```
        {
```

```
            index++;
```

```
            if(index >= temp.length() || temp.charAt(index) ==  
' ' || temp.charAt(index) == '.' || temp.charAt(index) == ',' || temp.charAt(index) ==  
'!' || temp.charAt(index) == ':' || temp.charAt(index) == ';' || temp.charAt(index) ==  
'?')
```

```
            {
```



```
        temp.insert(index,insertedText);
        index += lengthOfInsertedText;
    }
}

System.out.println(i+1 + ". " + temp);
}

System.out.print("\n");
}
```

### **Висновки**

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи с серіалізацією та десеріалізацією об'єктів .

Програма протестована, виконується без помилок.