

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту



Лабораторна робота N2

З дисципліни

«Алгоритмізація та програмування»

Виконав:

Студент групи КН-108

Жеребецький Олег

Викладач:

Грабовська Н.Р.

Львів – 2019р.

1.1 Розробник

Жеребецький Олег, КН-108, номер варіанту індивідуального завдання – 11.

Мета

- Набуття навичок розробки власних контейнерів.
- Використання ітераторів.
- Тривале зберігання та відновлення стану об'єктів.
- Ознайомлення з принципами серіалізації/десеріалізації об'єктів.
- Використання бібліотек класів користувача.

Вимоги

1. Розробити клас-контейнер, що ітерується для збереження початкових даних Вашого варіанту завдання з попередньої роботи (Прикладні задачі. Список з 1-15 варіантів) у вигляді масиву рядків з можливістю додавання, видалення і зміни елементів.

2. В контейнері реалізувати та продемонструвати наступні методи:

- `String toString()` повертає вміст контейнера у вигляді рядка;
- `void add(String string)` додає вказаний елемент до кінця контейнеру;
- `void clear()` видаляє всі елементи з контейнеру;
- `boolean remove(String string)` видаляє перший випадок вказаного елемента з контейнера;
- `Object[] toArray()` повертає масив, що містить всі елементи у контейнері;
- `int size()` повертає кількість елементів у контейнері;
- `boolean contains(String string)` повертає `true` , якщо контейнер містить вказаний елемент;
- `boolean containsAll(Container container)` повертає `true` , якщо контейнер містить всі елементи з зазначеного у параметрах;

○ `public Iterator<String> iterator()` повертає ітератор відповідно до `Interface Iterable` . 3. В класі ітератора відповідно до `Interface Iterator` реалізувати методи:

○ `public boolean hasNext()` ;

○ `public String next()` ;

○ `public void remove()` .

4. Продемонструвати роботу ітератора за допомогою циклів `while` и `for each`

.

5. Забороняється використання контейнерів (колекцій) і алгоритмів з `Java Collections Framework` .

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

7. Обмінятися відкомпільованим (без початкового коду) службовим класом (`Utility Class`) рішення одного варіанту задачі (Прикладні задачі. Список з 1-15 варіантів) з сусіднім номером. 1 міняється з 2, 2 з 3, 3 з 4, 4 з 5 і т.д. Останній, 15 міняється з 1 варіантом і далі аналогічно.

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

9. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.

10. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

1.2 Задача

Реалізувати контейнер для даних першої лабораторної та можливість серіалізації та десеріалізації.

2 Опис програми

Дана розроблена програма дозволяє занести у файл дані , які згодом можна буде звідти видобути. А також вносити декілька даних за рахунок контейнера та ітерації.

2.1 Засоби ООП

Для виконання завдання були перезаписані абстрактні класи ітератора. Та використані для обробки дій з контейнером.

2.2 Ієрархія та структура класів

1. Клас Main, який містить 1 функцію – main.
2. Клас Container – саме контейнер зі своїми методами для запису даних та їх виделення та виводу і обігу розміру даних + описаний окремо клас ітератора з оверайднутими методами некст та хезнекст.
3. Клас Ser_De_Ser – клас щоб перевести дані у формат серіалізації та навпаки.
4. Клас SER_DE – клас уже для серіалізації та десеріалізації, тобто роботі з файлом.
5. Клас List – клас для виконання роботи то вводу даних узгоджених із завданням у першій лабораторній.
6. Клас Menu – для обробки даних введених з консолі для розпізнавання команд з консолі та розпізнавання дебагу та допомоги при потребі.

2.3 Найважливіші фрагменти коду

Серіалізація та десеріалізація

File Edit Source Refactor Navigate Search Project Run Window Help

Container.java List.java Menu.java Ser_De_Ser.java SER_DE.java Main.java

```
38     int i = point_next;
39     if (!hasNext())
40         throw new NoSuchElementException();
41     String[] array = Container.this.Arra_text;
42     if (i >= array.length)
43         throw new ConcurrentModificationException();
44     point_next = i + 1;
45     return Arra_text[last_out = i] + " // " + Arra_list[i] + " // " + Arra_lenght[i];
46 }
47
48 ///////////////////////////////////////////////////
49 @Override
50 public void remove() {
51     if (last_out < 0)
52         throw new IllegalStateException();
53
54     try
55     {
56         Container.this.remove(last_out);
57         point_next = last_out;
58         last_out = -1;
59     } catch (IndexOutOfBoundsException ex)
60     {
61         throw new ConcurrentModificationException();
62     }
63 }
64
65 //.....
66
67
68 public void add(int index, String text, String list, int lengh){
69     if (index>Arra_text.length){
70         addSize();
71     }
72     Arra_text[index]=text;
73     Arra_list[index]=list;
74 }
```

Problems Javadoc Declaration Console

<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (19 xsir, 2019 p, 22:49:23)

Writeable Smart Insert 52 : 51

File Edit Source Refactor Navigate Search Project Run Window Help

Container.java List.java Menu.java Ser_De_Ser.java SER_DE.java Main.java

```
74     Arra_list[index]=list;
75     Arra_lenght[index]=length;
76     if (index>size)
77     {
78         size=index;
79     }
80
81 ///////////////////////////////////////////////////
82 public int size()
83 {
84     return size;
85 }
86
87 ///////////////////////////////////////////////////
88 public void out()
89 {
90     Iterator<String> Iter =iterator();
91     for(String str : Arra_text )
92     {
93         if(Iter.hasNext())
94             System.out.println(Iter.next());
95         else break;
96     }
97 }
98
99 //.....
100
101 public boolean contains(String text)
102 {
103     int index=-1;
104     for (int i =0;i<size;i++){
105         if (text.equals(Arra_text[i])){
106             index=i;
107             break;
108         }
109     }
110 }
```

Problems Javadoc Declaration Console

<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (19 xsir, 2019 p, 22:49:23)

Writeable Smart Insert 52 : 51

Ппор - oleh_zherebetskiy/src/ua/lpnuai/oo/Zherebetskiy02/Container.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Container.java List.java Menu.java Ser_De_Ser.java SER_DE.java Main.java

```
110 }
111 }
112 if (index== -1)
113     return false;
114     return true;
115 }
116 //////////////////////////////////////////////////
117 public boolean containsAll(Container<E> con1)
118 {
119
120     if (Arra_text.equals(con1))
121     {
122         return false ;
123     }
124     return true;
125 }
126 //////////////////////////////////////////////////
127 public void add(String text, String list , int length){
128     if (size>Arra_text.length){
129         addSize();
130     }
131
132     Arra_text[size]=text;
133     Arra_list[size]=list;
134     Arra_lenght[size]=length;
135     size++;
136 }
137
138 //////////////////////////////////////////////////
139 private void addSize(){
140     String[] arra_w = new String[Arra_text.length*2];
141     System.arraycopy(Arra_text,0,arra_w,0,size-1);
142     Arra_text = arra_w;
143 }
144
145 //////////////////////////////////////////////////
146
```

Problems Javadoc Declaration Console

<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (19 ksir, 2019 p., 22:49:23)

Writeable Smart Insert 52 : 51

Ппор - oleh_zherebetskiy/src/ua/lpnuai/oo/Zherebetskiy02/Container.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

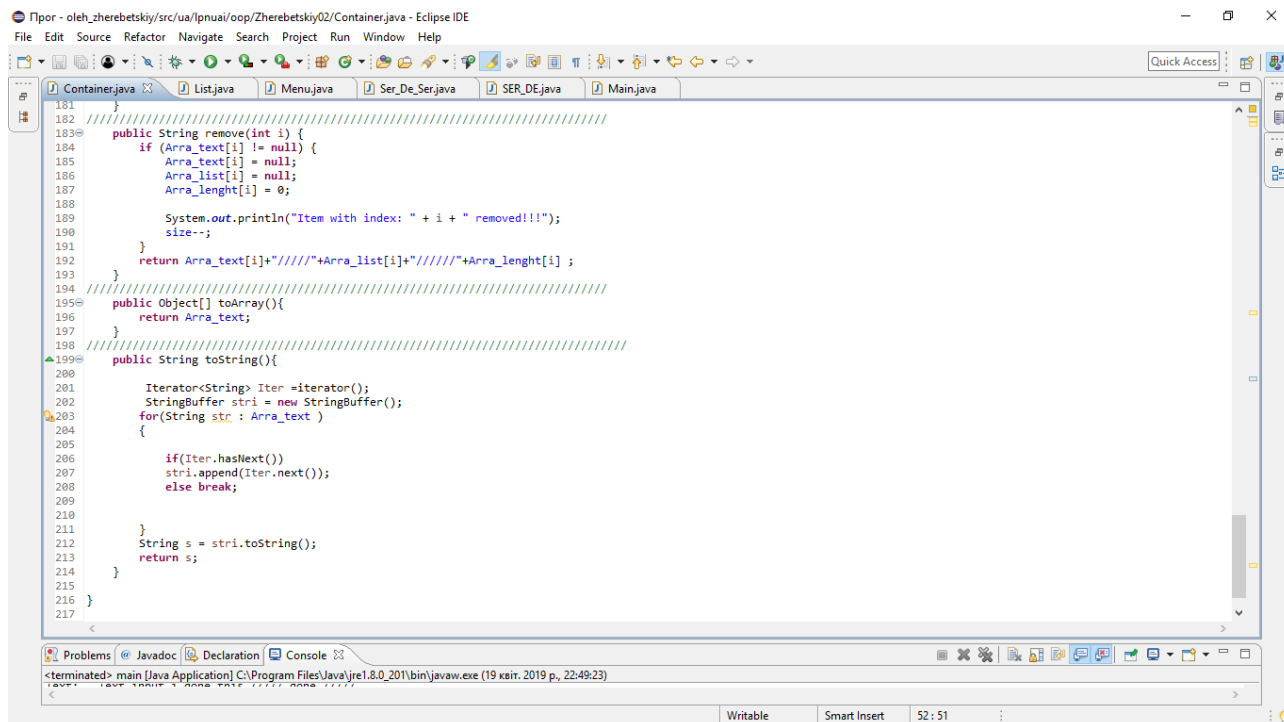
Container.java List.java Menu.java Ser_De_Ser.java SER_DE.java Main.java

```
146 //////////////////////////////////////////////////
147 public void clear(){
148     for (int i =0;i<size;i++){
149
150         Arra_text[i]=null;
151         Arra_list[size]=null;
152         Arra_lenght[size]=0;
153     }
154     size=0;
155 }
156 //////////////////////////////////////////////////
157 public boolean remove(String text){
158     int index=-1;
159     for (int i =0;i<size;i++){
160         if (text.equals(Arra_text[i])){
161             index=i;
162             break;
163         }
164     }
165     if (index== -1)
166         return false;
167
168     for (int i =index;i<size;i++){
169         Arra_text[i]=Arra_text[i+1];
170         Arra_list[i]=Arra_list[i+1];
171         Arra_lenght[i]=Arra_lenght[i+1];
172     }
173
174     Arra_text[size-1]=null;
175     Arra_list[size-1]=null;
176     Arra_lenght[size-1]=0;
177
178     size--;
179     return true;
180 }
181
182 //////////////////////////////////////////////////
183
```

Problems Javadoc Declaration Console

<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (19 ksir, 2019 p., 22:49:23)

Writeable Smart Insert 52 : 51



Робота в консолі


```

..
.
.
Con  sol comand:      -a
.
Input text:      Text input i do this
.
Input line:      done
.
Word lenght:      2
..
.
.
Con  sol comand:      -ca
..
.
.
Con  sol comand:      -sf
..
.
.
Con  sol comand:      -of
Text input i do this ///// done ///// 2
..
.
.
Con  sol comand:      -oc
Text input i do this ///// done ///// 2
Text input i do this ///// done ///// 2 ///// 0
..
.
.
Con  sol comand:      -os
Text input i do this ///// done ///// 2Text input i do this ///// done ///// 2 ///// 0
..

```

```

.
Con  sol comand:      -cl
..
.
.
Con  sol comand:      -os
..
.
.
Con  sol comand:      -sz
0
..
.
.
Con  sol comand:      -d
Text:  Text input i do this ///// done ///// 2
..
.
.
Con  sol comand:      -b
Text:  Text input i do this ///// done ///// 2
List:  done
Word lenght:      2
..
.
.
Con  sol comand:      -c
Add to text end sign (" \0")

Variable for moveing srought ( i = 0 ), word length (num1 = 0), calculation mod point = 0

point: 1          num1: 1

```

```
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0
Text:  Text input i done this ///// done /////

Text:  Text input i done this ///// done /////

Text:  Text input i done this ///// done /////

num1:  0      End of word

i:      39      -char number

point: 1      num1: 1

i:      40      -char number

point: 0      End of word)

Text:  Text input i done this ///// done ///// 2

Text:  Text input i done this ///// done ///// 2

num1:  0      End of word

i:      41      -char number
.
Work done!
Text:  Text input i done this ///// done ///// 2
List:  done
Word lenght:  2

..
.
.
Con  sol comand:  -e
.
```

3. Варіанти використання

Дана програма може використовуватись для багаторазової обробки різних текстів щодо виправлення неправильних слів певної довжини на інші.

ВИСНОВКИ

У ході роботи розвинулись навички написання серіалізації та десеріалізації а також розуміння ітератора та `for each`. Також реалізований контейнер накопичення декількох даних + можливість команд з боку користувача.