**Лабораторна робота №5**

**Розробка власних контейнерів. Ітератори**

**Мета:** Набуття навичок розробки власних контейнерів. Використання ітераторів.

**1 ВИМОГИ**

1. Розробити клас-контейнер, що [ітерується](http://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html) для збереження початкових даних завдання л.р. №3 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](http://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html).
3. В класі ітератора відповідно до [Interface Iterator](http://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html) 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](https://docs.oracle.com/javase/8/docs/technotes/guides/collections/).
   1. **Розробник**

* Дубина Валерій Валерійович
* Группа: КІТ-119д
* Варіант: 8

1. **ОПИС ПРОГРАМИ**
   1. **Було використано наступні засоби:**

string.length() – визначення довжини рядка;

Iterator<String> iterator – ітератор.

* 1. **Ієрархія та структура класів**

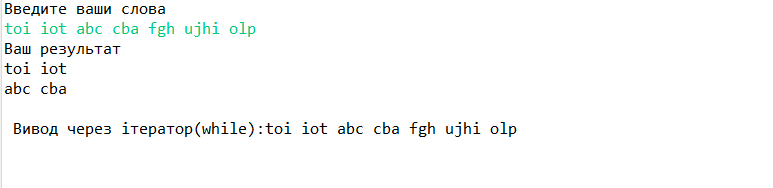
Було створено 4 класи:

* public class Main – містить метод main;
* public class Container – клас, що містить методи для роботи з контейнером;
* public inteface Iterator – клас, що містить методи для роботи з ітератором.
* public class UtilityClass- клас, що виконує основне завдання
  1. **Важливі фрагменти програми**

1. **int** size=0;
2. String[] arr=**new** String[255];
3. **public** String toString()
4. {
5. **if** (size ==-1) {
6. System.***out***.print("\nМасив пустий. Елементів немає. Попернуто null");
7. **return** **null**;
8. }
9. String temp=**new** String();
10. **for** (**int** i=0;i<size;i++)
11. temp+=arr[i];
12. **return** temp;
13. }
14. **void** add(String string)
15. {
16. String newArr[] = **new** String[size + 1];
17. **for** (**int** i = 0; i < size; i++)
18. newArr[i] = arr[i];
19. newArr[size] = string;
20. size++;
21. arr = newArr;
22. }
23. **void** clear()
24. {
25. **while** (size!=0)
26. iterator().remove();
28. }
29. **boolean** remove (String string)
30. {
32. **if** (size == 0) **return** **false**;
33. **for** (**int** i=0;i<size;i++)
35. **if** (arr[i].equals(string+" "))
37. { **for** (; i < size-1; i++)
38. arr[i]=arr[i+1];
39. **this**.arr[--size] = **null**;
40. **return** **true**;
41. }
43. **return** **false**;
45. }
47. **int** size()
48. {
49. **return** size;
50. }
51. **boolean** contains(String string)
52. {
53. **if** (size ==0) **return** **false**;
54. **for** (**int** i=0;i<size;i++)
55. **if** (arr[i]==string)
56. **return** **true**;
58. **return** **false**;
59. }
60. Object[] toArray()
61. {
62. **return** arr;
63. }
64. **boolean** containsAll(Container container)
65. {
66. **if**(container.size==size)
67. **if**(container.arr==arr)
68. **return** **true**;
69. **return** **false**;

72. }

**Результати роботи програми**



**Висновок**

Під час виконання лабораторної роботи було набуто навичок розробки власних контейнерів та роботи з ітераторами у середовищі Eclipse IDE.