Науменко Д.

КІТ 101.8а

Лабораторна робота №5. Розробка власних контейнерів. Ітератори

**Мета** :

-Набуття навичок розробки власних контейнерів.

-Використання ітераторів.

**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](https://oop-khpi.github.io/" \l "task_03_app)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/).

**Код програми** :

funcs.java

package lab5;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Iterator;

public class funcs implements Iterable{

private ArrayList<String> str;

public funcs(ArrayList<String> str) {

this.str = str;

}

public ArrayList<String> getData() {

return str;

}

public void setData(ArrayList<String> str) {

this.str = str;

}

public void add(String string) {

str.add(string);

System.out.println("Added Word:" + string);

System.out.println("New Str: "+ str);

}

public void clear() {

str.clear();

System.out.println("Clear: " + str);

}

public boolean remove(String string) {

str.add("noise ");

str.add("warning");

str.add(string);

System.out.println("Str before removing: " + str);

return str.remove(string);

}

public String toString()

{

StringBuilder build = new StringBuilder();

for(String iter1: str)

{

build.append(iter1);

}

System.out.println("Str without previous word: " + build);

return build.toString();

}

public Object[] toArray() {

String[] mas = str.toArray(new String[str.size()]);

System.out.println("Array: " + Arrays.toString(mas));

return str.toArray();

}

public int size() {

int s = str.size();

System.out.println("size: " + s);

return s;

}

public Iterator<String> iterator() {

return new newIterator();

}

private class newIterator<String> implements Iterator{

int index;

@Override

public boolean hasNext() {

if(index < str.size()) {

return true;

}

return false;

}

@Override

public Object next() {

return str.get(index++);

}

}

public boolean contains(String string) {

newIterator iterator = new newIterator();

while(iterator.hasNext()) {

if(iterator.next().toString().equals(string)) {

return true;

}

}

return false;

}

public boolean containsAll(funcs container) {

return this.str.equals(container.str);

}

}

interf.java

package lab5;

public interface interf {

public boolean hasNext();

public String next();

}

Iterable.java

package lab5;

import java.util.Iterator;

public interface Iterable {

String toString();

void add(String string);

void clear();

boolean remove(String string);

Object[] toArray();

int size();

boolean contains(String string);

boolean containsAll(funcs container);

public Iterator<String> iterator();

}

main.java

package lab5;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Iterator;

public class main {

public static void main(String[] args) {

ArrayList<String> arr = new ArrayList<String>(Arrays.asList("level", "level", "oppo"));

funcs str = new funcs(arr);

Iterator<String> iterator = str.iterator();

String str1 = "oppo";

String str2="high";

str.add(str1);

str.clear();

str.remove(str2);

str.contains("look");

str.toString();

str.toArray();

while(iterator.hasNext()) {

System.out.println("Method Next: " + iterator.next());

}

str.size();

}

}

**Робота програми** :

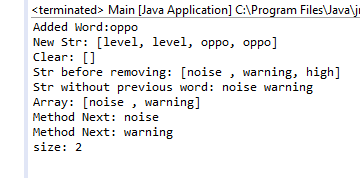


Рис.1. – Робота програми

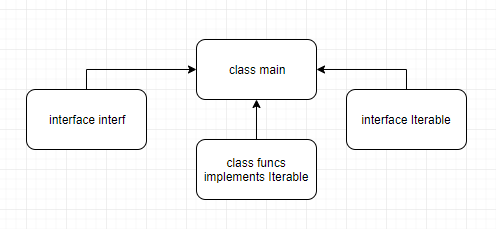
**Ієрархія класів** :

Рис. 2 – Ієрархія класів.

**Висновок** : на лабораторній роботі ми набули навичок розробки власних контейнерів.