ЛАБОРАТОРНА РОБОТА №5

РОЗРОБКА ВЛАСНИХ КОНТЕЙНЕРІВ. ІТЕРАТОРИ

*Мета:*

Набуття навичок розробки власних контейнерів. Використання ітераторів.

1 ВИМОГИ

* 1. Розробник
* Соколенко Дмитро Григорович
* група КІТ 102.8а
  1. Загальне завдання

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/#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/).

2 ОПИС ПРОГРАМИ

**2.1 Опис логічної структури**

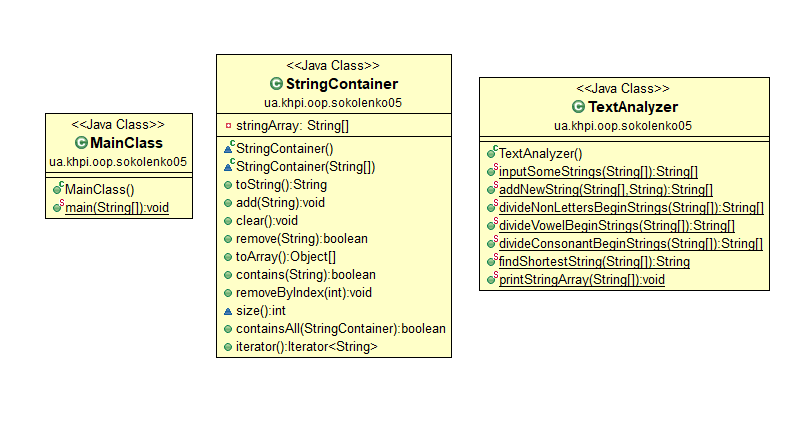


Рис 1 – Діаграма класів

**2.2 Фрагменти коду**

Частина класу контейнеру:

**public** **class** StringContainer **implements** Iterable<String> {

**private** String [] stringArray;

StringContainer() {

stringArray = **new** String[0];

}

StringContainer(String[] newStringArray) {

**if**(newStringArray == **null**) {

stringArray = **new** String[0];

}

**else** {

stringArray = newStringArray;

}

}

**public** String toString() {

String result = **new** String();

**for** (**int** index = 0; index < stringArray.length; index++) {

result += ("| " + index +" - " + stringArray[index] + " ");

}

result += "|";

**return** result;

}

**public** **void** add(String addedString) {

String[] newArray = **new** String[stringArray.length + 1];

System.*arraycopy*(stringArray, 0, newArray, 0, stringArray.length);

newArray[stringArray.length] = addedString;

stringArray = newArray;

}

**public** **void** clear() {

stringArray = **new** String[0];

}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

**3.1 Ілюстрація роботи програми**

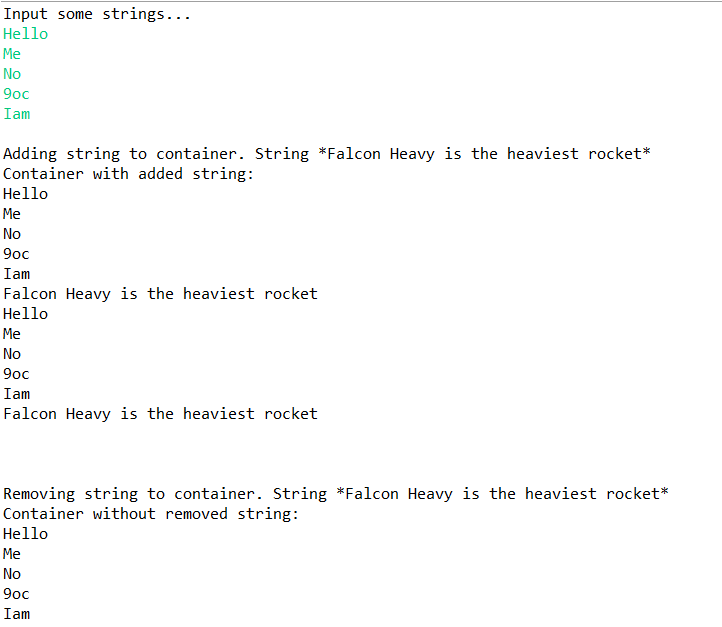


Рис 2 – результат роботи у меню

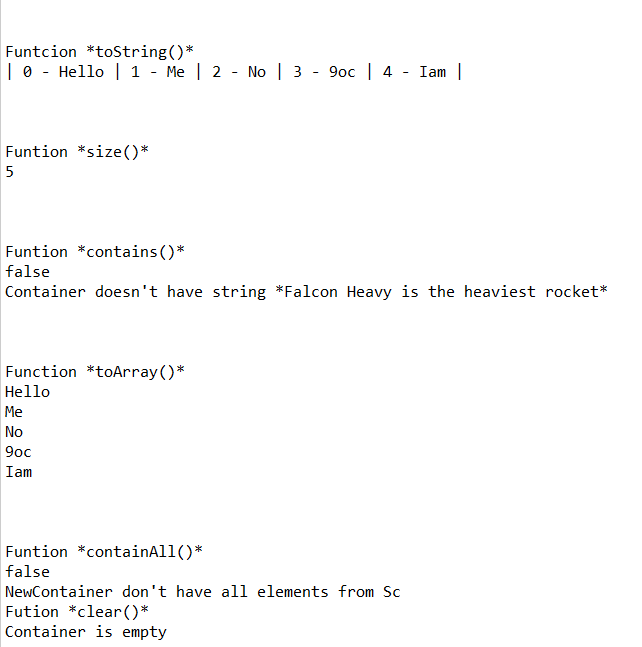


Рис 3 – результат роботи у меню (продовження)

**ВИСНОВОК**

Отримали навички з реалізації діалогового режиму роботи з користувачем в консольних програмах мовою Java.