5. Розробка власних контейнерів. Ітератори

Мета: Набуття навичок розробки власних ітераторів. Використання ітераторів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кулик Данііл Ігорович
- НТУ “ХПІ” 1.КІТ102.8а
- Варіант 11

1.2 Загальне завдання

1. Розробити клас-контейнер, що ітерується 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.
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*.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі відсутні об’єктно-орієнтовані методи.

2.2 Ієрархія та структура даних

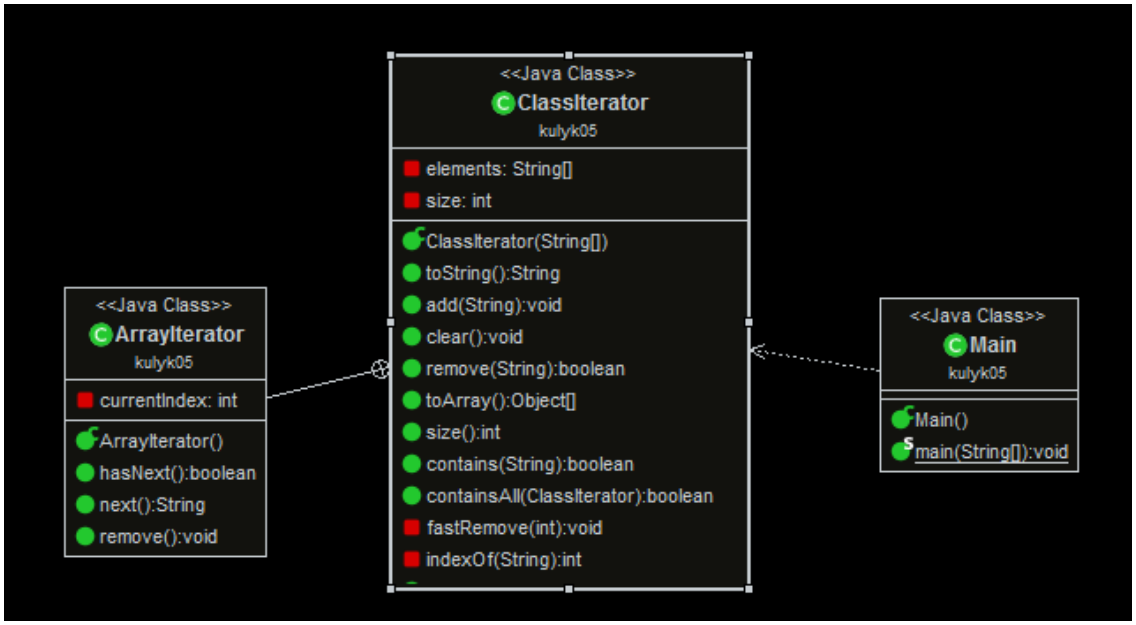

Рисунок 1 – діаграма класів

2.3 Важливі фрагменти програми

```
94          public class ArrayIterator implements Iterator<String>{
95                  private int currentIndex = 0;
96
97          @Override
98          public boolean hasNext() {
99              return currentIndex < size && elements[currentIndex] != null;
100         }
101
102         @Override
103         public String next() {
104             return elements[currentIndex++];
105         }
106
107         @Override
108         public void remove() {
109             throw new UnsupportedOperationException();
110         }
111          }
112
113     @Override
114     public Iterator<String> iterator() {
115             return new ArrayIterator();
116     }
117 }
```

Рисунок 2 – тіло класу ArrayIterator

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма створена задля демонстрації використання та тестування різноманітних методів розробленого контейнера, що містить масив рядків.

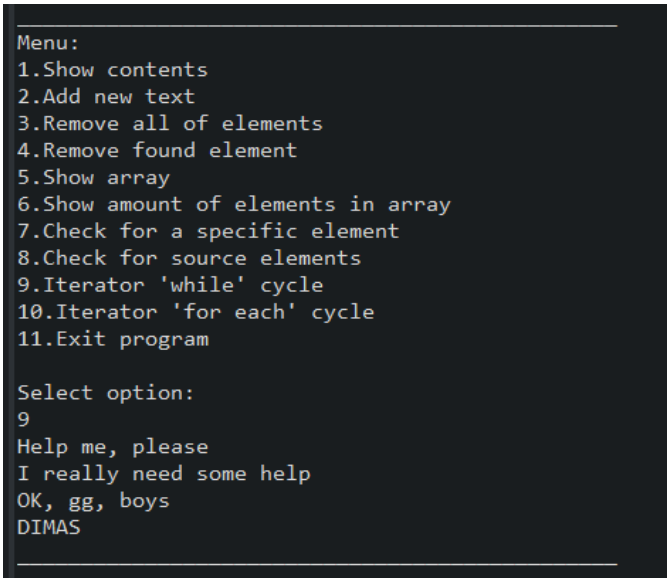

Рисунок 3 – Результати роботи циклу *for each*

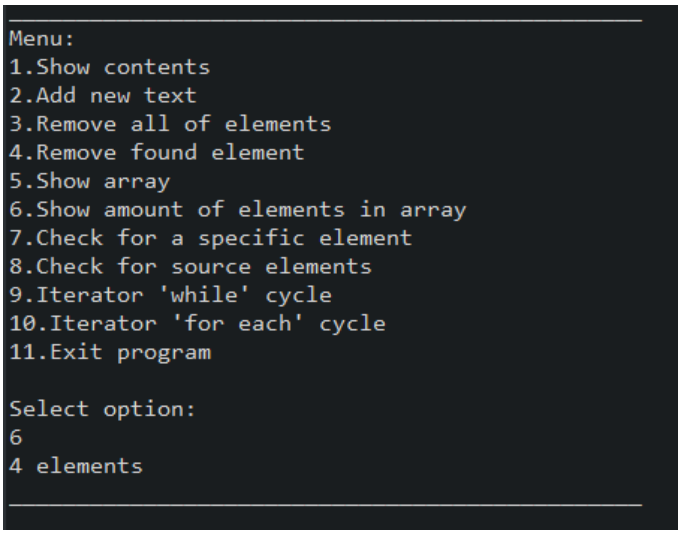

Рисунок 4 – Результати роботи методу *size()*

ВИСНОВКИ

В даній лабораторній роботі розробив та використав власний ітератор задля зберігання і обробки елементів контейнера.