

ЛАБОРАТОРНА РОБОТА №5
РОЗРОБКА ВЛАСНИХ КОНТЕЙНЕРІВ. ІТЕРАТОРИ

Мета: Набуття навичок розробки власних контейнерів. Використання ітераторів.

1 ВИМОГИ

1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ 102.8(а);
- 19-12-2019р..

1.2 Загальне завдання

Розробити клас-контейнер, що ітерується для збереження початкових даних завдання л.р. №3 у вигляді масиву рядків з можливістю додавання, видалення і зміни елементів. 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`.

В класі ітератора відповідно до `Interface Iterator` реалізувати методи:

`public boolean hasNext();`

`public String next();`

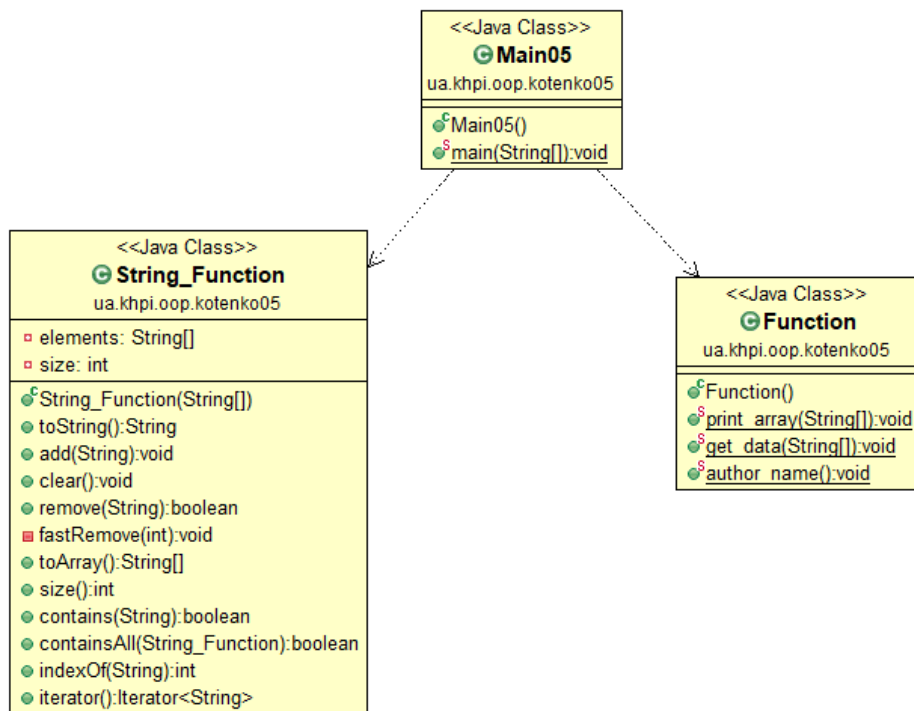
`public void remove().`

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

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

2 ОПИС ПРОГРАМИ

2.1 Опис логічної структури



2.2 Фрагменти коду

```
public void add(String string) {
    if (size == elements.length){
        String[] newArray = new String[elements.length + 1];
        System.arraycopy(elements, 0, newArray, 0, size);
        elements = newArray;
    }
    elements[size] = string;
    size++;
}
```

Рисунок 2.1 – Додавання елементу

```
public void clear() {
    for (int i = 0; i < size; i++) {
        elements[i] = null;
    }
    size = 0;
}
```

Рисунок 2.2 – Очистка массиву

```

public boolean remove(String string) {
    if (string == null) {
        for (int index = 0; index < size; index++)
            if (elements[index] == null) {
                fastRemove(index);
                return true;
            }
    } else {
        for (int index = 0; index < size; index++)
            if (string.equals(elements[index])) {
                fastRemove(index);
                return true;
            }
    }
    return false;
}

```

Рисунок 2.3 – Видалення елементу

```

public String toString() {
    String result = " ";
    for (int i = 0; i < size; i++) {
        result += (elements[i] + " ");
    }
    return result;
}

```

Рисунок 2.4 – Занесення до строки

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Ілюстрація роботи програми

```

Enter some names(5)
Sergey
Dima
Roma
Luck
Vlad
false
Sergey Dima Roma Luck Vlad
-1

Sergey
Dima
Roma
Luck
Vlad

```

Рисунок 3.1 – Введена інформація з відпрацьованими функціями

ВИСНОВОК

В інтегрованому середовищі *Eclipse* розроблена програма мовою Java. Виконання програми дозволяє продемонструвати роботу контейнерів та ітераторів.