

# 1. Тема роботи

Розробка власних контейнерів. Ітератори. Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача

## 1. ВИМОГИ

### 1.1 Розробник

Подоба Арсен Мирославович  
КН-108  
Варіант 1

### 1.2 Загальне завдання

#### Вимоги

1. Розробити клас-контейнер, що ітерується для збереження початкових даних Вашого варіанту завдання з попередньої роботи (Прикладні задачі. Список з 1-15 варіантів) 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`

5. Забороняється використання контейнерів (колекцій) і алгоритмів з Java Collections Framework .
6. Реалізувати і продемонструвати тривале зберігання/відновлення розробленого контейнера за допомогою серіалізації/десеріалізації .
7. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення одного варіанту задачі (Прикладні задачі. Список з 1-15 варіантів) з сусіднім номером. 1 міняється з 2, 2 з 3, 3 з 4, 4 з 5 і т.д. Останній, 15 міняється з 1 варіантом і далі аналогічно.
8. Продемонструвати послідовну та вибірккову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
9. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
10. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

## 2. ОПИС ПРОГРАМИ

### 2.1 Засоби ООП

Декомпозиція для розділення завдання між класами, інкапсуляція, імплементація

### 2.2 Важливі фрагменти програми

Обробка даних користувача

```
import java.util.Iterator;

public interface MyList <T> extends Iterable <T>
{
    String toString();

    void add (String string);

    String get (int index);

    void set (String string, int index);

    void clear();

    boolean remove(String string);

    int size();

    boolean contains (String string);

    String[] toArray();
}
```

```
Iterator<T> iterator();  
  
boolean containsAll(MyArrayList myArrayList);  
}
```

## ВИСНОВКИ

В ході лабораторної роботи, я розробив власний контейнер для зберігання рядків. Навчився працювати з ітераторами. Навчився використовувати серіалізацію об'єктів та відновлення їх через десеріалізацію.