

# Тема роботи

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

## Мета

- Набуття навичок розробки власних контейнерів.
  - Використання ітераторів.
- Тривале зберігання та відновлення стану об'єктів.
- Ознайомлення з принципами серіалізації/десеріалізації об'єктів.
- Використання бібліотек класів користувача.

## 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 foreach.

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.

### Важливі фрагменти програми

```
public class InnerIterator implements Iterator<Integer> {
    int index;

    public InnerIterator() { index = 0; }

    public boolean hasNext() { return index < size; }

    public Integer next() { return data[index++]; }

    public void remove() { throw new UnsupportedOperationException(); }
}

public Iterator<Integer> iterator() { return new InnerIterator(); }
```

```
public class MyCo implements Iterable<Integer>, Serializable {

    private int[] data;
    private int size = 0;

    public MyCo() { data = new int[0]; }
}
```

## Приклад використання програми

```
Enter data:
asd asad asd

Commands:
a: Enter text
b: View text
c: Serialize
d: Deserialize
e: Delete
q: Quit
```

```
b

*****
asd asad asd
vowels: 4, consonants: 7
*****
```

```
*****
asd asad asd
vowels: 4, consonants: 7
*****
fgh gfiaeirkuu geirug
vowels: 9, consonants: 10
*****
```