

Кафедра систем штучного інтелекту

**Лабораторна робота №9
Розробка власних контейнерів. Ітератори.
Серіалізація/десеріалізація об’єктів. Бібліотека
класів користувача
з дисципліни
«Об’єктно-орієнтоване програмування»**

Виконала:
студентка групи КН-108
Гетьман Соломії
Викладач:
Гасько Р. Т.

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.

Вимоги

- 1) Розробити клас-контейнер, що ітерується;
- 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;
- 3) Пропредемонструвати роботу ітератора за допомогою циклів while і for each.
- 4) Реалізувати і продемонструвати тривале зберігання/відновлення розробленого контейнера за допомогою серіалізації/десеріалізації

Код програми

```
package ua.lpnuai.oop.hetman09;

import java.util.Iterator;

public class Main {
    public static void main(String[] args) {
        String[] arr = {"1", "2", "3", "4", "5", "6", "7", "8" , "9", "10",
"11"};
        String[] arr_3 = {"1", "2", "3"};
        NotArrayList list = new NotArrayList(arr);
        System.out.println(list);
        list.remove("3");
        list.remove("5");
        list.remove("3");
        System.out.println(list);
        String[] arr_2 = list.toArray();
        System.out.println(list.size());
        System.out.println(list.contains("1"));
        System.out.println(list.contains("3"));
        System.out.println(list.containsAll(arr_3));
        Iterator it = list.iterator();
        for (; it.hasNext(); ) {
            System.out.println(it.next());
        }

        it.remove();

        while(it.hasNext()){
            System.out.println(it.next());
        }

        list.save(list);

        NotArrayList list2 = list.load();
```

```

        System.out.println(list2);

        String[] arr_for_each = list2.toArray();

        for(String a : arr_for_each){
            System.out.println(a);
        }
    }
}

package ua.lpnuai.oop.hetman09;

import java.io.*;
import java.util.Iterator;

public class NotArrayList implements Serializable {

    private String[] array = new String[10];
    private int index = 0;
    private int size = 10;

    public NotArrayList(String[] arr) {
        for (int i = 0; i < arr.length; i++) {
            if (enoughSpace(index + 1)) {
                this.add(arr[i]);
            } else {
                array = this.increaseSize(array);
                this.add(arr[i]);
            }
        }
    }

    public void add(String elem) {
        array[index++] = elem;
    }

    public void getList() {
        for (int i = 0; i < index; i++) {
            System.out.print(array[i] + " ");
        }
    }

    @Override
    public String toString() {
        StringBuilder string = new StringBuilder("[");
        for (int i = 0; i < index; i++) {
            if (i == index - 1) {
                string.append(array[i]);
            } else string.append(array[i] + "," + " ");
        }
        string.append("]");
        return string.toString();
    }

    private boolean enoughSpace(int index) {
        return index <= size;
    }

    private String[] increaseSize(String[] array) {
        size = (size * 3) / 2 + 1;
        String[] arr_temp = new String[size];
        System.arraycopy(array, 0, arr_temp, 0, array.length);
        return arr_temp;
    }

    public void clear() {

```

```

        array = new String[size];
        index = 0;
    }

    public boolean remove(String string) {
        for (int i = 0; i < index; i++) {
            if (array[i].equals(string)) {
                String[] arr_temp = new String[size];
                System.arraycopy(array, 0, arr_temp, 0, i);
                System.arraycopy(array, i + 1, arr_temp, i, index - i);
                index--;
                array = arr_temp;
                return true;
            }
        }
        return false;
    }

    public boolean remove(int it){
        if (it <= index) {
            String[] arr_temp = new String[size];
            System.arraycopy(array, 0, arr_temp, 0, it);
            System.arraycopy(array, it + 1, arr_temp, it, index - it);
            index--;
            array = arr_temp;
            return true;
        }
        else return false;
    }

    public String[] toArray() {
        return array;
    }

    public int size() {
        return index;
    }

    public boolean contains(String string) {
        for (int i = 0; i < index; i++) {
            if (array[i].equals(string)){
                return true;
            }
        }
        return false;
    }

    public boolean containsAll(String[] strings){
        int found = 0;
        for (int i = 0; i < index; i++){
            for(int j = 0; j < strings.length; j++){
                if (array[i].equals(strings[j])){
                    found++;
                }
            }
        }
        return found == strings.length;
    }

    public Iterator<String> iterator() {
        return new Itr();
    }

    public void save(Object object){
        try {
            FileOutputStream fos = new FileOutputStream("data.bin");
            ObjectOutputStream os = new ObjectOutputStream(fos);

```

```

        os.writeObject(object);
        os.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public NotArrayList load(){
    NotArrayList list = new NotArrayList(new String[]{" "});
    try {
        FileInputStream fis = new FileInputStream("data.bin");
        ObjectInputStream ois = new ObjectInputStream(fis);

        list = (NotArrayList) ois.readObject();

        ois.close();

    } catch (IOException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    return list;
}

private class Itr implements Iterator<String>{
    int next;
    int current = -1;

    Itr() {}

    public boolean hasNext(){
        return next != index;
    }

    public String next(){
        int i = next;
        next = i+1;
        String[] data = NotArrayList.this.array;
        return data[current = i];
    }

    public void remove(){
        NotArrayList.this.remove(current);
        next = 0;
        current = -1;
    }
}
}

```