

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №9**  
з дисципліни  
«ООП»

**Виконав:**  
студент групи КН-109  
Пилип А. Р.  
**Викладач:**  
Гасько Р.Т.

Львів – 2018

## Завдання

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

## Код програми

```
package com.pylyp.lpnu.lab9;

import com.pylyp.lpnu.lab9.Helper;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Scanner;

public class Mainn {

    public static void main(final String[] args) throws
FileNotFoundException, IOException, ClassNotFoundException
    {

        show();
        Scanner s = new Scanner(System.in);
        BufferedReader read = new BufferedReader(new
InputStreamReader(System.in));
        Container storage = new Container();

        int a=0;
        final int end=11;
        while(a != end)
        {
            a=s.nextInt();
            switch (a) {
                case 1:
                    System.out.println("Введіть новий
елемент:");
                    storage.add(read.readLine());
                    break;
                case 2:
                    if (storage.remove(read.readLine())) {
                        System.out.println("Елемент
видалиний!");
                    } else {
                        System.out.println("Такого елемента не
існує!");
                    }
                    break;
                case 3:
```

```

        storage.clear();
        System.out.println("Контейнер був
видалений!");
        break;
    case 4:
        System.out.println(storage.toString());
        break;
    case 5:
        storage.compare();
        break;
    case 6:
        storage.sort((String[]) storage.toArray());
        System.out.println(storage.toString());
        break;
    case 7:
        System.out.println("-----");
        Helper.PrintSymbols(storage.toString());

        Helper.PrintSymbolNumbers(storage.toString());
        break;
    case 8:
        System.out.print("Exit");
        break;
    default : break;
    }
}
s.close();
}

public static void show() {
    System.out.println("Меню");
    System.out.println("1. Додавання елемента");
    System.out.println("2. Видалення елемента");
    System.out.println("3. Очистка");
    System.out.println("4. Вивід");
    System.out.println("5. Порівняння");
    System.out.println("6. Сортування");
    System.out.println("7. Використання класу іншого
студента");
    System.out.println("8. Вихід");
}
}

```

```

package com.pylyp.lpnu.lab9;

```

```

public class Helper {
    public static void PrintSymbols(final String line) {

        StringBuilder str = new StringBuilder();

        for (char symbol : line.toCharArray()) {

```

```

        str.append(symbol + "\t");
    }

    System.out.println(str.toString());
}

public static void PrintSymbolNumbers(final String line) {

    StringBuffer str = new StringBuffer("");
    for(char ch : line.toCharArray()){
        if(Character.isAlphabetic(ch))
            str.append(String.format("%-3s", ch));
    }

    System.out.println(str.toString());
}
}

```

```

package com.pylyp.lpnu.lab9;

```

```

import java.util.Arrays;
import java.util.Comparator;
import java.util.Iterator;
import java.util.NoSuchElementException;

```

```

public class Container implements Iterable<String> {

```

```

    private static int el;
    private String[] arr = new String[el];

```

```

    @Override

```

```

    public String toString() {

```

```

        StringBuffer buff = new StringBuffer();

```

```

        for(int i = 0; i < el; i++) {
            buff.append(arr[i] + " ");
        }

```

```

        return buff.toString();
    }

```

```

    public void add(String string) {

```

```

        el++;

        final int size = arr.length;

        arr = Arrays.copyOf(arr, size + 1);

        arr[size] = string;
    }

    public void clear() {

        arr = new String[el];
    }

    public boolean remove(String string) {

        if(el == 0)
            return false;

        String[] del = new String[el];
        del = arr;

        int j;

        for (j = 0; j < el; j++) {

            if (del[j].equals(string))
                break;

            else if(j == el - 1)
                return false;
        }

        for (int k = j; k < el - 1; k++)
            del[k] = del[k + 1];

        el--;

        arr = new String[el];

        for(int i = 0; i < el; i++)
            arr[i] =del[i];

        return true;
    }

    public Object[] toArray() {

        return arr;
    }

```

```

public int size() {
    return el;
}

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

    return false;
}

public boolean containsAll(Container container) {
    for(int i = 0; i < el; i++) {
        if(container.arr[i].equals(arr[i]))
            return true;
    }

    return false;
}

public void sort(final String[] arr) {
    Arrays.sort(arr, new Comparator<String>() {
        public int compare(final String str1, final String
str2) {
            return
str1.toString().compareTo(str2.toString());
        }
    });
}

public String[] search(final int length) {
    String[] validStr = new String[arr.length];
    for (int i = 0; i < arr.length; i++) {
        if (arr[i].length() == length) {
            validStr[i] = arr[i];
        }
    }

    return validStr;
}

```

```

public void compare() {

    String equalElems = "";
    int countOfEqual = 0;

    for (int i = 0; i < arr.length; ++i) {

        for (int j = i + 1; j < arr.length; ++j) {

            if (arr[i].equals(arr[j])) {

                equalElems += arr[i];
                countOfEqual++;
            }
        }
    }

    if (equalElems.isEmpty()) {

        System.out.println("Немає однакових елементів");
    } else {

        countOfEqual++;
        System.out.println(equalElems + " - " +
countOfEqual);
    }
}

```

```

@Override
public ContainerIter<String> iterator() {
    return new ContainerIter<String>(arr);
}

```

```

@SuppressWarnings("hiding")
class ContainerIter<String> implements Iterator<String> {

    private int low;
    private int high;

    public ContainerIter(String[] array) {

        this.low = 0;
        this.high = array.length - 1;
        el = array.length;
    }

    @Override

```

```

public boolean hasNext() {
    return this.low <= this.high;
}

@Override
public String next() {
    int temp;

    if (!this.hasNext()) {
        throw new NoSuchElementException();
    }

    temp = low;
    low++;

    return (String) arr[temp];
}

@Override
public void remove() {
    if (low < high + 1) {
        arr = Arrays.copyOf(arr, el - 1);
    }

    el--;
    high--;
}
}
}

```

### Робота програми:

```

Меню
1. Додавання елемента
2. Видалення елемента
3. Очистка
4. Вивід
5. Порівняння
6. Сортювання
7. Використання класу іншого студента
8. Вихід
1
Введіть новий елемент:
5 7 9
4
5 7 9

```