Лабораторна робота №6

[**СЕРІАЛІЗАЦІЯ/ДЕСЕРІАЛІЗАЦІЯ ОБ'ЄКТІВ. БІБЛІОТЕКА КЛАСІВ КОРИСТУВАЧА**](https://oop-khpi.gitlab.io/#tasks)

**Мета:** Тривале зберігання та відновлення стану об'єктів. Ознайомлення з принципами серіалізації/десеріалізації об'єктів. Використання бібліотек класів користувача.

ВИМОГИ

**Розробник:**

- Чугунов Вадим Юрійович;

- КІТ-119а;

- Варіант №24.

**Загальне завдання:**

1. Реалізувати і продемонструвати тривале зберігання/відновлення [раніше розробленого контейнера](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task05/) за допомогою [серіалізації/десеріалізації](https://docs.oracle.com/javase/8/docs/technotes/guides/serialization/index.html).
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення [задачі л.р. №3](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/" \l "_4) з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
4. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
5. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

ОПИС ПРОГРАМИ

**Опис змінних**

Kontainer kontain = new Kontainer(); // об’єкт контейнера

Scanner scan = new Scanner(System.in); // змінна сканера

boolean loop = true; // змінна типу буліан для запуску циклу

int choose; // змінна для збереження числа

int choose2; // змінна для збереження числа

String str1; // змінна для збереження строки

String str2; // змінна для збереження строки

int temp = 0; //змінна для збереження тимчасових данних

FileOutputStream file = new FileOutputStream("tempFile.txt"); // змінна збереження даних

ObjectOutputStream object = new ObjectOutputStream(file); // обджект

**Ієрархія та структура класів**

**Class** Main – точка входу в програму;

**Class** Functions - допоміжний клас в якому реалізуються функції;

**Class** Kontainer implements Serializable //класс-контейнер з реалізованими функціями збереження та відновлення даних;

**Class** Helper – мій хелпер клас, який виконує завдання лабораторної роботи №3;

**Class** ZavadskiyHelper – хелпер клас одногрупника, який ми імпортували до нашої програми.

ТЕКСТ ПРОГРАМИ

**Текст файлу Main.java**

package ua.oop.khpi.chugunov06;

import ua.oop.khpi.chugunov03.Helper;

import ua.oop.khpi.zavadskiyHelper.ZavadskiyHelper;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.Scanner;

public class Main {

/\*\*

\* An entry point of program

\* @param args

\*/

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

Kontainer kontain = new Kontainer();

Scanner scan = new Scanner(System.in);

boolean loop = true;

int choose;

int choose2;

String str1;

String str2;

int temp = 0;

while (loop) {

Functions.chooseMenu();

choose = scan.nextInt();

switch (choose) {

case 1:

str1 = Functions.initializeStr();

kontain.aDDBack(str1);

break;

case 2:

System.out.println("Your container:");

kontain.showArray();

break;

case 3:

System.out.println("\nSorting alphabetically:");

temp = kontain.getSize();

System.out.println("Enter number from 1 to " + temp);

choose2 = scan.nextInt();

kontain.alphabeticalSorting(choose2 - 1);

break;

case 4:

kontain.clearArray();

System.out.println("Clear container successfully!");

break;

case 5:

System.out.println("Container to String: ");

System.out.println(kontain.arrayToStr());

break;

case 6:

str2 = Functions.initializeStr();

if (kontain.remove(str2)) {

System.out.println("Successfully");

break;

} else {

System.out.println("This string is absent");

break;

}

case 7:

str2 = Functions.initializeStr();

if (kontain.contains(str2)) {

System.out.println("This string contains in container");

break;

} else {

System.out.println("This string is absent");

break;

}

case 8:

System.out.println("\nRunning helper method of another student...");

ZavadskiyHelper.start(

new StringBuilder(kontain.arrayToStr()));

break;

case 9:

System.out.println("\nRunning MY helper method...");

String myTaskString = kontain.arrayToStr();

String [] array = Helper.SplitString(myTaskString);

Helper.PrintResult(array);

System.out.println();

break;

case 10:

str2 = Functions.initializeStr();

int temper = kontain.search(str2);

if (temper == 0) {

System.out.println("This element is not abs");

break;

} else {

System.out.print("This elements contains in index: " + (temper - 1));

break;

}

case 11:

if(kontain.compression()) {

System.out.println("Elements are equal");

break;

} else {

System.out.println("Elements are not equal");

break;

}

case 12:

System.out.println("Saving to file");

FileOutputStream file = new FileOutputStream("tempFile.txt");

ObjectOutputStream object = new ObjectOutputStream(file);

object.writeObject(kontain);

object.close();

System.out.println("Successfully");

break;

case 13:

System.out.println("Read from file");

FileInputStream fileInput = new FileInputStream("tempFile.txt");

ObjectInputStream objectInput = new ObjectInputStream(fileInput);

kontain = (Kontainer) objectInput.readObject();

objectInput.close();

System.out.println("Successfully");

break;

case 14:

System.out.println("\n\nThanks for working!");

loop = false;

break;

default:

System.out.println("Error. Mistake number!");

break;

}

}

}

}

**Текст файлу Functions.java**

package ua.oop.khpi.chugunov06;

import java.util.Arrays;

import java.util.Scanner;

final class Functions {

static void chooseMenu() {

System.out.println("\n\nChoose the right variant:");

System.out.println("1. Create and add string in container");

System.out.println("2. Show elements in container");

System.out.println("3. Sort elements in container");

System.out.println("4. Clear container");

System.out.println("5. Show container to string");

System.out.println("6. Deleting element from container");

System.out.println("7. Check contains in container");

System.out.println("8. Helper class another student");

System.out.println("9. My Helper class");

System.out.println("10. Search elements in container");

System.out.println("11. Compression two elements in container");

System.out.println("12. Save elements in file");

System.out.println("13. Read elements from file");

System.out.println("14. The End");

System.out.print("Write:");

}

private static String[] differentWords(final String str) {

int word = 0;

int begin = 0;

int count = 0;

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

if (str.charAt(i) == ' ') {

count++;

}

}

String[] wordArr = new String[++count];

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

if (str.charAt(i) == ' ') {

wordArr[word] = str.substring(begin, i);

word++;

begin = i + 1;

}

}

wordArr[word] = str.substring(begin, str.length());

return wordArr;

}

static String initializeStr() {

Scanner scan = new Scanner(System.in);

System.out.print("Enter string: ");

return scan.nextLine();

}

static String sortAlphabetical(final String str) {

System.out.println("\nSorting by alphabetical now");

String str1 = "";

String[] wordArr = differentWords(str);

Arrays.sort(wordArr);

str1 = reload(wordArr);

System.out.println("Successfully");

return str1;

}

private static String reload(final String[] str2) {

StringBuilder str1 = new StringBuilder();

for (String s : str2) {

str1.append(s);

str1.append(" ");

}

str1.deleteCharAt(str1.length() - 1);

return str1.toString();

}

}

**Текст файлу Kontainer.java**

package ua.oop.khpi.chugunov06;

import java.io.Serializable;

import java.util.Arrays;

import java.util.Scanner;

import static ua.oop.khpi.chugunov06.Functions.sortAlphabetical;

public class Kontainer implements Serializable {

/\*\*

\* First size for array.

\*/

private final int size = 50;

/\*\*

\* Array contains all data.

\*/

private String[] array = new String[size];

/\*\*

\* Counter of number elements.

\*/

private int count = 0;

void showArray() {

if (count == 0) {

System.out.println("Empty mass");

} else {

System.out.println();

for (int i = 0; i < count; i++) {

System.out.println(array[i]);

}

}

}

void aDDBack(final String str1) {

if (count == array.length) {

array = Arrays.copyOf(array, array.length \* 2);

array[count++] = str1;

} else {

array[count++] = str1;

}

}

/\*\*

\* The override to add method for adding an elem of string array.

\* @param str - string array

\*/

public void addElemOfArray(final String[] str) {

for (String i : str) {

this.aDDBack(i);

}

}

String arrayToStr() {

StringBuilder str1 = new StringBuilder("");

if (count != 0) {

str1 = new StringBuilder(array[0]);

str1.append(" ");

for (int i = 1; i < count; i++) {

str1.append(array[i]);

str1.append(" ");

}

}

return str1.toString();

}

void clearArray() {

for (int i = 0; i < count; i++) {

array[i] = null;

}

count = 0;

}

int getSize() {

return count;

}

boolean contains(final String str) {

boolean cont = false;

for (int i = 0; i < count; i++) {

if (cont) {

return cont;

} else {

String str1;

str1 = array[i];

cont = str.equals(str1);

if (i == count - 1) {

if (!cont) {

return cont;

}

}

}

}

return cont;

}

boolean remove(final String str) {

boolean remov = false;

int coun = 0;

for (int i = 0; i < count; i++) {

if (remov) {

break;

} else {

remov = str.equals(array[i]);

coun++;

}

}

array[coun-1] = null;

int temp = coun-1;

for (int i = coun; i < count; i++) {

array[temp++] = array[i];

}

count--;

return remov;

}

public Object[] toArray() {

if (array == null) {

return null;

}

return Arrays.copyOf(array, count);

}

private String elementByIndex(final int index) {

return array[index];

}

void alphabeticalSorting(final int index) {

array[index] = sortAlphabetical(array[index]);

}

boolean containsAll(final Kontainer container) {

boolean result;

if (container.getSize() != count) {

return false;

}

for (int i = 0; i < count; i++) {

result = array[i].equals(container.elementByIndex(i));

if (!result) {

return false;

}

}

return true;

}

int search(final String str) {

boolean cont = false;

int temp = 0;

for (int i = 0; i < count; i++) {

temp++;

if (cont) {

return temp;

} else {

String str1;

str1 = array[i];

cont = str.equals(str1);

if (i == count - 1) {

if (!cont) {

System.out.println("This element is abs");

return 0;

}

}

}

}

return temp;

}

boolean compression() {

Scanner scan = new Scanner(System.in);

System.out.println("Enter index first string: ");

int choose1 = scan.nextInt();

System.out.println("Enter index second string: ");

int choose2 = scan.nextInt();

return array[choose1-1].equals(array[choose2-1]);

}

}

**Текст класу Helper.java**

public class Helper {

public static String[] SplitString(String text) {

List<String> words = new ArrayList<>();

StringBuilder builder = new StringBuilder();

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

if((int)symbol == 32 | (int)symbol == 33 |(int)symbol == 58|(int)symbol == 44|(int)symbol == 46) {

words.add(builder.toString());

builder = new StringBuilder();

continue;

}

builder.append(symbol);

}

if(builder.length() != 0) {

words.add(builder.toString());

}

for (int i = 0; i < words.size(); i++) {

if(words.get(i).length() == 0) {

words.remove(i);

}

}

String[] output = new String[words.size()];

for (int i = 0; i < words.size(); i++) {

output[i] = words.get(i);

}

return output;

}

public static void PrintResult (String[]words){

System.out.println( "==========================");

System.out.println( "A Word" + "\t\t\t" + "Count");

System.out.println( "==========================");

HashMap<String, Integer> wordToCount = new HashMap<>();

for (String word : words) {

if (!wordToCount.containsKey(word)) {

wordToCount.put(word, 0);

}

wordToCount.put(word, wordToCount.get(word) + 1);

}

for (String word : wordToCount.keySet()) {

System.out.println(word + "\t\t\t" + wordToCount.get(word));

}

System.out.println( "==========================");

}

}

**Текст класу ZavadskiyHelper.java**

public class ZavadskiyHelper {

private static int mean = 0;

public static void start(StringBuilder ln) {

if(ln == null) {

throw new NullPointerException("line is null");

}

clottingSpace(ln);

ArrayList<StringBuilder> line = findArray(ln);

task(line);

}

static private ArrayList<StringBuilder> findArray(StringBuilder ln){

ArrayList<StringBuilder> line = new ArrayList<StringBuilder>();

ArrayList<Integer> IndexSpace = new ArrayList<Integer>();

int length = ln.length();

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

if(ln.charAt(i)==' ')

IndexSpace.add(i);

}

if(IndexSpace.size() == 0) {

line.add(new StringBuilder(ln));

}

else

{

line.add(new StringBuilder(ln.substring(0, IndexSpace.get(0))));

IndexSpace.add(length);

for(int i=1;i<IndexSpace.size();i++) {

if(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))!="")

line.add(new StringBuilder(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))));

}

}

return line;

}

static private StringBuilder clottingSpace(StringBuilder line) {

for(int i = 0; i< line.length()-1;i++)

{

if(line.charAt(i)==' '&&line.charAt(i+1)==' ') {

line.deleteCharAt(i+1);

i--;

}

}

if(line.charAt(line.length()-1)==' ')

line.deleteCharAt(line.length()-1);

if(line.charAt(0)==' ')

line.deleteCharAt(0);

System.out.println(line);

return line;

}

static private void task(ArrayList<StringBuilder> line)

{

int size =0;

int count;

for (StringBuilder strB: line) {

size += strB.length();

}

count = size;

int[] array\_numbers = new int [count];

int i=0;

for (StringBuilder strB : line) {

size = strB.length();

char[] arr = new char[size];

strB.getChars(0, size,arr,0);

for (char el : arr) {

if((int)el >=97 && (int)el<= 122)

{

array\_numbers[i] = (int)el;

array\_numbers[i] -=96;

}

else if((int)el >=65 && (int)el<= 90)

{

array\_numbers[i] = (int)el;

array\_numbers[i] -=64;

}

i++;

}

}

i=0;

char[] arrALL = new char[count];

for (StringBuilder strB : line) {

size = strB.length();

strB.getChars(0, size, arrALL, i);

i+=size ;

}

for (char c : arrALL) {

System.out.print(c+" ");

}

System.out.println("\n");

for (int c : array\_numbers) {

System.out.print(c+" ");

}

}

}

**РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ**

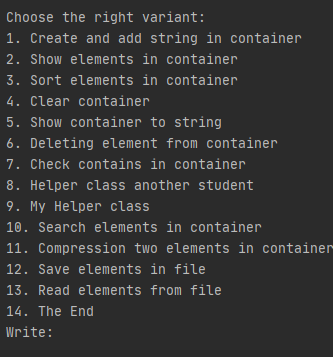


Рис. 6.1 – Результат роботи програми

a) б)

Рис. 6.2 – Результат роботи програми

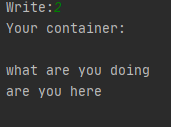
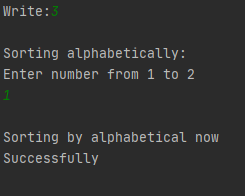
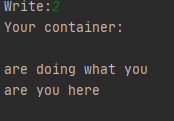
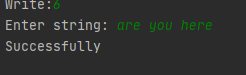
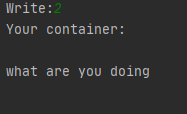
****

Рис. 6.3 – Результат роботи програми

1. б)

Рис. 6.4 – Результат роботи програми

** **

а) б)

Рис. 6.5 – Результат роботи програми

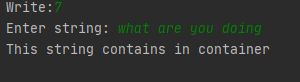


Рис. 6.6 – Результат роботи програми

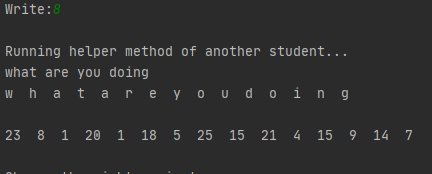
****

Рис. 6.7 – Результат роботи програми

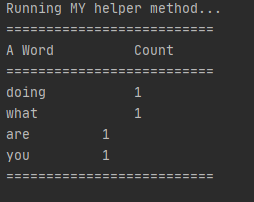
****

Рис. 6.8 – Результат роботи програми

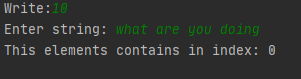
****

Рис. 6.9 – Результат роботи програми

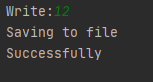
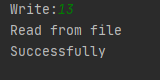
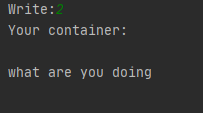
****

Рис. 6.10 – Результат роботи програми

** **

а) б)

Рис. 6.11 – Результат роботи програми

**ВАРІАНТИ ВИКОРИСТАННЯ**

Програму можна використовувати задля форматованої роботи з рядками. А саме сортування за алфавітом. Також для вибору доступно багато інших можливостей. Використовувати інтерактивне меню задля простого доступу до елементів програми.

**ВИСНОВОК**

Під час виконання лабораторної роботи було набуто навичок тривалого зберігання та відновлення стану об'єктів. Ознайомилися з принципами серіалізації/десеріалізації об'єктів.