Звіт

Автор: Васильченко С., 1.КІТ101.8б

Дата: 07.11.2019

Лабораторна робота №6

СЕРІАЛІЗАЦІЯ/ДЕСЕРІАЛІЗАЦІЯ ОБ’ЄКТІВ. БІБЛІОТЕКА КЛАСІВ КОРИСТУВАЧА

***Мета:***

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

**Вимоги:**

1. Реалізувати і продемонструвати тривале зберігання та відновлення раніше розробленого контейнера за допомогою серіалізації та десеріалізації.
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 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.

ЗАВДАННЯ ДО РОБОТИ

Реалізувати і продемонструвати тривале зберігання / відновлення раніше розробленого контейнера за допомогою серіалізації / десеріалізації.

ОПИС ПРОГРАМИ

***2.1 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

***2.2 Ієрархія та структура класів.***

final class Lab06 – головний клас. Містить метод main(точку входу у програму).

В класі використовується додаткова функція kontain.aDDBack(String str), яка відповідає за додання строки у контейнер.

public class StringLengthSort implements Comparator<String> - клас, який використовується для порівняння двох рядків, для подальшого сортування за довжиною.

class Functions - допоміжний клас в якому реалізуються функції

class ArgsCMD – класс-обробник командної строки

class Kontainer implements Serializable //класс-контейнер з реалізованими функціями збереження та відновлення даних

class StrBuilder // класс-хелпер друга

class UI // класс-хелпер друга

**ТЕКСТ ПРОГРАМИ**

File Lab06.java:

package sergey.lab05;

import sergey.lab05.Kontainer.IteratorMine;

final class Lab05 {

private Lab05() {

//default constructor

}

package sergey.Lab06;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.Scanner;

final class Lab06 {

private Lab06() {

//default constructor

}

/\*\*

\* @param args - ggg.

\* @throws IOException -ggg.

\* @throws ClassNotFoundException -ggg.

\*/

public static void main(final 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);

System.out.println("Successfully!");

break;

case 4:

System.out.println("\nSort by length: ");

temp = kontain.getSize();

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

choose2 = scan.nextInt();

kontain.lengthSorting(choose2 - 1);

System.out.println("Successfully!");

break;

case 5:

kontain.clearArray();

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

break;

case 6:

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

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

break;

case 7:

str2 = Functions.initializeStr();

if (kontain.remove(str2)) {

System.out.println("Successfully");

break;

} else {

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

break;

}

case 8:

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 9:

kontain.helperSanchez();

break;

case 10:

System.out.println("Sorting elements in container");

kontain.sortContainer();

System.out.println("Successfully");

break;

case 11:

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 12:

if(kontain.compression()) {

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

break;

} else {

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

break;

}

case 13:

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 14:

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

FileInputStream fileInput = new FileInputStream("tempFile");

ObjectInputStream objectInput = new ObjectInputStream(fileInput);

kontain = (Kontainer) objectInput.readObject();

objectInput.close();

System.out.println("Successfully");

break;

case 0:

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

loop = false;

break;

default:

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

break;

}

}

}

}

Functions.java:

package sergey.Lab06;

import sergey.lab05.StringLengthSort;

import java.util.Arrays;

import java.util.Scanner;

final class Functions {

private Functions() {

//default constructor

}

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. Sorting element alphabetically");

System.out.println("4. Sort element by length");

System.out.println("5. Clear container");

System.out.println("6. Show container to string");

System.out.println("7. Deleting element from container");

System.out.println("8. Check contains in container");

System.out.println("9. Execute string");

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

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

System.out.println("12. Compression two elements in container");

System.out.println("13. Save elements in file");

System.out.println("14. Read elements from file");

System.out.println("0. 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) {

String[] wordArr = differentWords(str);

Arrays.sort(wordArr);

return reload(wordArr);

}

static String sortByLength(final String str) {

String[] wordArr = differentWords(str);

StringLengthSort sort1 = new StringLengthSort();

Arrays.sort(wordArr, sort1);

return reload(wordArr);

}

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();

}

}

StringLengthSort.java:

package Lab4;

import java.util.Comparator;

public class StringLengthSort implements Comparator<String> {

@Override

public int compare(String o1, String o2) {

if (o1.length() > o2.length()) {

return 1;

} else {

if (o1.length() < o2.length())

return -1;

else

return 0;

}

}

}

Kontainer.java:

package sergey.Lab06;

import java.io.Serializable;

import java.util.Arrays;

import java.util.Scanner;

import static sergey.Lab06.Functions.sortAlphabetical;

import static sergey.Lab06.Functions.sortByLength;

/\*\*

\* Contain elements in program.

\*/

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;

}

}

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 lengthSorting(final int index) {

array[index] = sortByLength(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;

}

void helperSanchez() {

Scanner scan = new Scanner(System.in);

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

int temp = count;

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

int choose2 = scan.nextInt();

String temp1 = array[choose2];

array[choose2] = StrBuilder.execute(temp1, UI.getWord(), UI.getSentence());

}

void sortContainer() {

String[] newString;

newString = Arrays.copyOf(array, count);

StringLengthSort str1 = new StringLengthSort();

Arrays.sort(newString, str1);

array = Arrays.copyOf(newString, count);

}

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]);

}

}

UI.java:

package sergey.Lab06;

import java.util.Scanner;

class UI {

static int choice;

private static Scanner scan = new Scanner(System.in);

static void mainMenu() {

System.out.format("%n1. Enter values.%n");

System.out.format("2. Print values.%n");

System.out.format("3. Task completion.%n");

System.out.format("4. Print result.%n");

System.out.format("0. Exit.%n");

System.out.format("Enter your choose: ");

}

static void printInfo(final String text, final String word, final String sentence) {

System.out.format("%nText: " + text + "%n");

System.out.format("Word: " + word + "%n");

System.out.format("Sentence: " + sentence + "%n");

}

static void printResult(final String text, final String result) {

System.out.format("%nSource text: " + text + "%n");

System.out.format("Result: " + result + "%n");

}

static void getChoice() {

choice = scan.nextInt();

scan.nextLine();

}

static String getText() {

System.out.format("%nEnter text: ");

return scan.nextLine();

}

static String getWord() {

System.out.format("Enter word: ");

return scan.nextLine();

}

static String getSentence() {

System.out.format("Enter sentence: ");

return scan.nextLine();

}

}

StrBuilder.java:

package sergey.Lab06;

class StrBuilder {

static String execute(String text, String word, String sentence) {

System.out.println("\nText: " + text);

System.out.println("Word: " + word);

System.out.println("Sentence: " + sentence);

StringBuilder builder = new StringBuilder(text);

int index = 0;

while (true) {

index = builder.indexOf(word, index);

if (index < 0)

break;

builder.insert(index + word.length(), sentence);

index += word.length();

}

return builder.toString();

}

}

**ВАРІАНТИ ВИКОРИСТАННЯ**

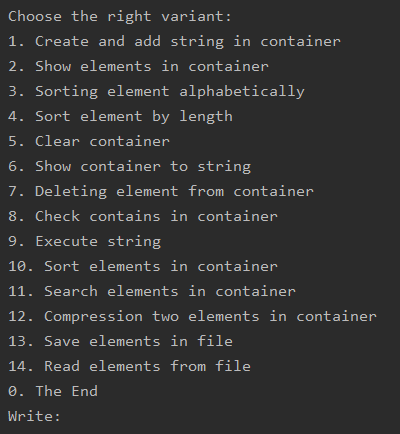
****

Рис. 6.1 – Результат роботи програми

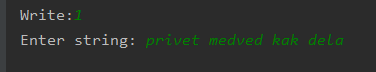
****

Рис. 6.2 – Результат роботи програми

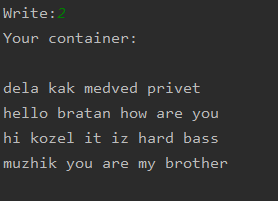
****

Рис. 6.3 – Результат роботи програми

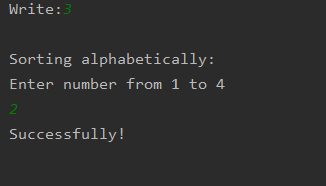
****

Рис. 6.4 – Результат роботи програми

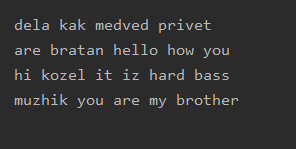
****

Рис. 6.5 – Результат роботи програми

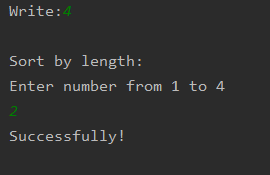


Рис. 6.6 – Результат роботи програми

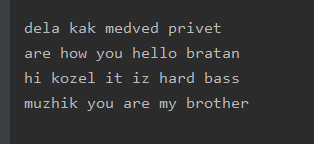


Рис. 6.7 – Результат роботи програми

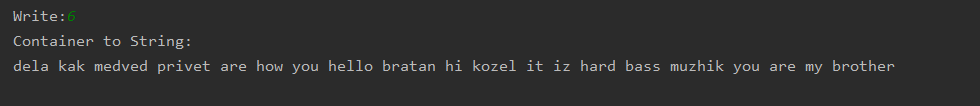


Рис. 6.8 – Результат роботи програми

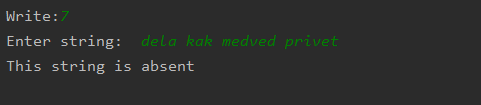


Рис. 6.9 – Результат роботи програми

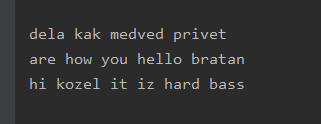


Рис. 6.10 – Результат роботи програми

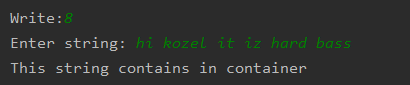


Рис. 6.11 – Результат роботи програми

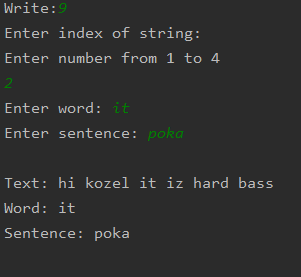


Рис. 6.12 – Результат роботи програми

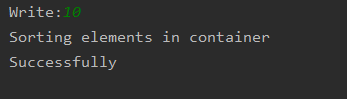


Рис. 6.13 – Результат роботи програми

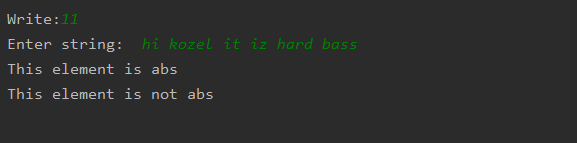


Рис. 6.14 – Результат роботи програми

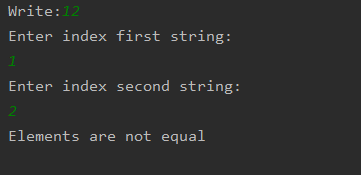


Рис. 6.15 – Результат роботи програми

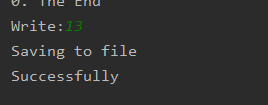


Рис. 6.16 – Результат роботи програми

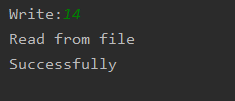


Рис. 6.17 – Результат роботи програми

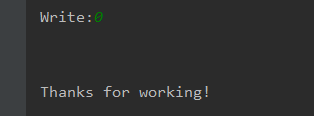


Рис. 6.18 – Результат роботи програми

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

**ВИСНОВКИ**

При виконанні лабораторної роботи набуто практичних навичок щодо реалізації тривалого зберігання та відновлення даних за допомогою серіалізації та десеріалізації. Завдання виконано. Програма працює успішно!