Звіт

Автор: Васильченко С., 1.КІТ101.8б

Дата: 07.11.2019

Лабораторна робота №5

РОЗРОБКА ВЛАСНИХ КОНТЕЙНЕРІВ. ІТЕРАТОРИ

***Мета:***

* Набуття навичок розробки власних контейнерів.
* Використання ітераторів.

**Вимоги:**

1. Розробити клас-контейнер, що ітерується для збереження початкових даних завдання л.р. №3 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.

1. В класі ітератора відповідно до Interface Iterator реалізувати методи:

* public boolean hasNext();
* public String next();
* public void remove().

1. Продемонструвати роботу ітератора за допомогою циклів while и for each.
2. Забороняється використання контейнерів (колекцій) і алгоритмів з Java Collections Framework.

ЗАВДАННЯ ДО РОБОТИ

Ввести декілька рядків. Упорядкувати, а потім вивести рядки за алфавітом (перший пріоритет) та в порядку зростання їх довжини (другий пріоритет).

ОПИС ПРОГРАМИ

***2.1 Опис змінних:***

String[] str1 //масив строк

ArgsCMD Arguments //обработка аргументів командного рядка

Scanner scan //сканер водимих даних

boolean loop //для запуску цикла

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

IteratorMine iter = (IteratorMine) kontain.iterator(); // ітератор для переміщення по елементам контейнера

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

final class Lab05 – 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 Iterable<String> //класс-контейнер з реалізованими функціями збереження та відновлення даних

**ТЕКСТ ПРОГРАМИ**

File Lab05.java:

package sergey.lab05;

import sergey.lab05.Kontainer.IteratorMine;

final class Lab05 {

private Lab05() {

//default constructor

}

/\*\*

\* Entry point in program.

\* @param args - arguments of main method.

\*/

public static void main(final String[] args) {

Kontainer kontain = new Kontainer();

String str1;

str1 = Functions.initializeStr();

Functions.showString(str1);

String str2 = Functions.sortAlphabetical(str1);

System.out.println();

Functions.showString(str2);

kontain.aDDBack(str1);

kontain.aDDBack(str2);

kontain.aDDBack(str1);

kontain.aDDBack(str2);

kontain.aDDBack(str1);

kontain.aDDBack(str2);

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

kontain.showArray();

kontain.lengthSorting(2);

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

kontain.showArray();

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

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

kontain.alphabeticalSorting(1);

System.out.println("\nWrite with iterator: ");

IteratorMine iter = (IteratorMine) kontain.iterator();

for (String s : kontain) {

System.out.println(s);

}

iter.next();

iter.remove();

System.out.println();

kontain.showArray();

}

}

Functions.java:

package sergey.lab05;

import java.util.Arrays;

import java.util.Scanner;

final class Functions {

private Functions() {

//default constructor

}

public static void chooseMenu() {

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

System.out.println("1. Create string");

System.out.println("2. Show string");

System.out.println("3. Sorting alphabetically");

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

System.out.println("5. Delete string");

System.out.println("0. The End");

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

}

static void showString(final String str) {

System.out.print(str);

}

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

int w = 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[w] = str.substring(begin, i);

w++;

begin = i + 1;

}

}

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

return wordArr;

}

public static String initializeStr() {

Scanner scan = new Scanner(System.in);

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

String str1;

str1 = scan.nextLine();

return str1;

}

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;

}

static String sortByLength(final String str) {

String str1 = "";

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

String[] wordArr = differentWords(str);

StringLengthSort sort1 = new StringLengthSort();

Arrays.sort(wordArr, sort1);

str1 = reload(wordArr);

System.out.println("Successfully");

return str1;

}

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

StringBuilder str1 = new StringBuilder("");

str1 = new StringBuilder(str2[0]);

str1.append(" ");

for (int i = 1; i < str2.length; i++) {

str1.append(str2[i]);

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;

}

}

}

**ВАРІАНТИ ВИКОРИСТАННЯ**

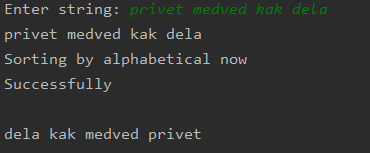
****

Рис. 5.1 – Результат роботи програми

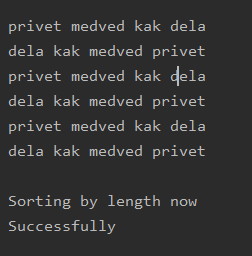
****

Рис. 5.2 – Результат роботи програми

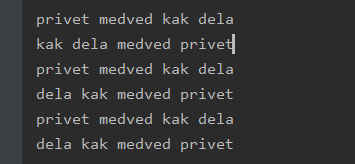
****

Рис. 5.3 – Результат роботи програми

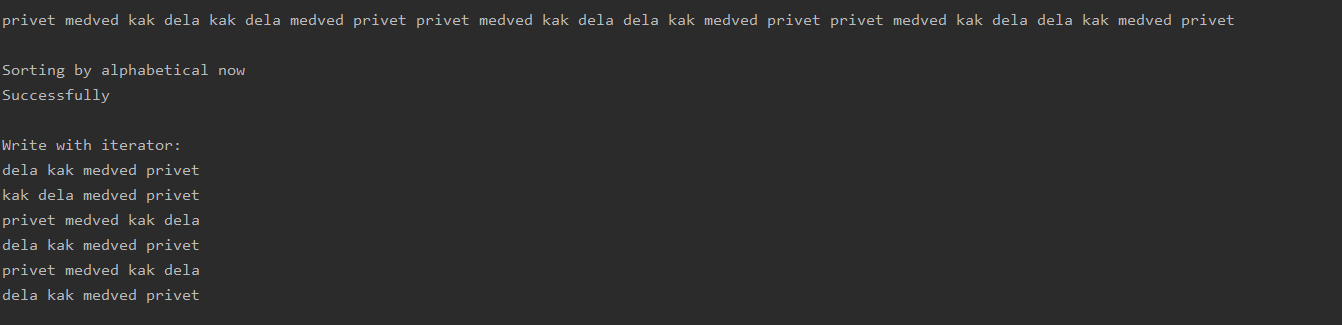
****

Рис. 5.4 – Результат роботи програми

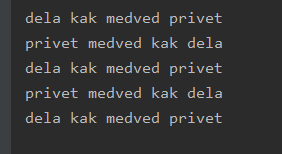
****

Рис. 5.5 – Результат роботи програми

Програму можна використовувати задля форматованої роботи з рядками. А саме сортування за алфавітом та за довжиною слова. Використовувати інтерактивне меню задля простого доступу до елементів програми.

**ВИСНОВКИ**

Завдяки даній лабораторній роботі було закріплено знання щодо розробки власних утилітарних класів. Було набуто навички, щодо використання рядків та масивів. Було реалізовано діалоговий режим роботи з користувачем консольних програм, набуто практичні навички розробки власних контейнерів, використання ітераторів. Завдання виконано. Програма працює успішно!