Розробник:

Огонькова Наталія Максимівна

Група КІТ 101.8б

Варіант №11

**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.
3. ОПИС ПРОГРАМИ
   1. Опис змінних

String str3 = null; //слово, що вводить користувач

**int** word = 0; //змінна для підрахунку кількості слів

**int** begin\_of\_word = 0; //початок

**int** number = 5; //розмір слова, яке змінеться

String str1 = “Hello my friend.”; //текст, що вводить користувач

String str2; //змінна для тексту

**int** count = 0; //кількість відступів

1.2 Ієрархія та структура класів

**class** Lab05 – точка входу в програму

**class** Func – клас, що містить функції

**class** Container – клас, в якому містяться контейнери

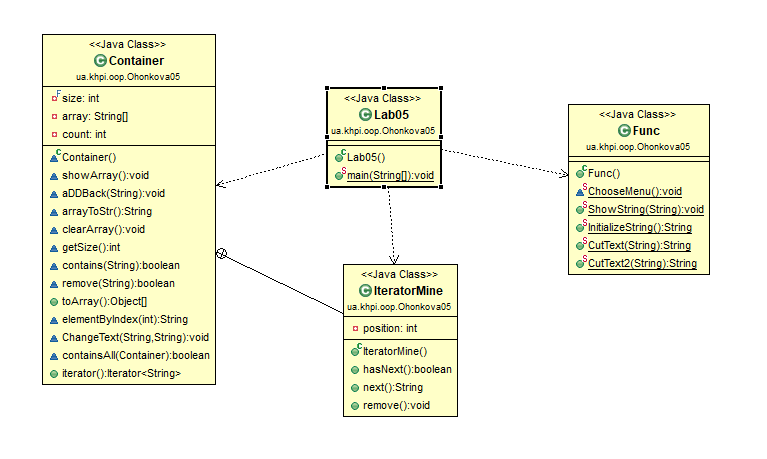


Рисунок 1 – Класи програми

* 1. Текст програми

**class** Lab05

package ua.khpi.oop.Ohonkova05;

import java.util.Arrays;

import java.util.Scanner;

import ua.khpi.oop.Ohonkova05.Container.IteratorMine;

import static ua.khpi.oop.Ohonkova04.Func.\*;

public class Lab05 {

/\*\*

\* @author NATALIA

\* @version 4.0

\* Here start point of the program

\* Function in which menu actions are performed

\* until the user presses exit program does not end

\*

\* \*/

public static void main(String[] args) {

Container container = new Container();

String str3;

str3 = Func.InitializeString();

Func.ShowString(str3);

String str1 = "Hello my friend.";

System.out.println();

container.aDDBack(str1);

container.aDDBack(str1);

container.aDDBack(str1);

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

System.out.println("Show Array: ");

container.showArray();

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

container.ChangeText(str1, str3);

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

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

IteratorMine iter = (IteratorMine) container.iterator();//перемещаться по элементам

for (String s : container) {

System.out.println(s);

}

iter.next();

iter.remove();

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

System.out.println("Show new Array: ");

container.showArray();

int size = container.getSize();

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

System.out.println("Size of Array: " + size);

}

}

**class** Func

**package** ua.khpi.oop.Ohonkova05;

**import** java.util.Scanner;

/\*

\* class Function in which all used functions are stored

\*

\* @author NATALIA

\* @version 4.2

\* Here start point of the program

\* \*/

**public** **class** Func {

/\*

\* Function that show user point of menu that he can choose

\* \*/

**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. Split text");

System.***out***.println("4. Counting now");

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

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

}

/\*

\* Function show our string that we wrote

\* @param str is our string

\* \*/

**public** **static** **void** ShowString(String str){

System.***out***.println("Your string:");

System.***out***.println(str);

}

/\*

\* Function reads input string

\* @return str1 as our used string

\* \*/

**public** **static** String InitializeString(){

Scanner scan = **new** Scanner(System.***in***);

System.***out***.print("Enter your word: ");

String str1 = **new** String();

str1 = scan.nextLine();

**return** str1;

}

/\*

\* Function truncates a string to a point and show the first sentence

\* @param index1 looking for dot

\* @return str

\* \*/

**public** **static** String CutText (String str) {

**int** index1=str.indexOf('.');

str = str.substring(index1+1);

**return** str;

}

/\*

\* Function truncates a string to a point and show the second sentence

\* @param index1 looking for dot

\* @return str

\* \*/

**public** **static** String CutText2 (String str) {

**int** index1=str.indexOf('.');

str = str.substring(0, index1+1);

**return** str;

}

}

**class** Container

package ua.khpi.oop.Ohonkova05;

import java.util.Arrays;

import java.util.Iterator;

import java.util.NoSuchElementException;

/\*\*

\* Contain elements in program.

\*/

class Container implements Iterable<String> {

/\*\*

\* First size for array.

\*/

private final int size = 10;

/\*\*

\* Array contains all data.

\*/

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

/\*\*

\* Counter of number elements.

\*/

private int count = 0;

/\*\*

\* Showing array`s data.

\*/

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]);

}

}

}

/\*\*

\* First size for array.

\* @param str1 -

\*/

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() {

array = 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) {

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++;

if (i == count - 1) {

break;

}

}

}

array[coun - 1] = null;

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

array[coun - 1] = array[coun++];

}

count--;

return remov;

}

public Object[] toArray() {

if (array == null) {

return null;

}

return Arrays.copyOf(array, count);

}

String elementByIndex(final int index) {

return array[index];

}

void ChangeText (String str1, String str3) {

int word = 0;

int begin\_of\_word = 0;

int count = 0;

int number = 5;

String str2 = null;

for (int i = 0; i < str1.length(); i++) {

if (str1.charAt(i) == ' ') {

count++;

}

}

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

for (int i = 0; i<str1.length();i++) {

if(str1.charAt(i)==' ') {

wordArr[word] = str1.substring(begin\_of\_word,i);

word++;

begin\_of\_word = i+1;

}

}

wordArr[word] = str1.substring(begin\_of\_word);

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

str2 = wordArr[i];

if (str2.length() == number) {

wordArr[i] = str3;

}

}

StringBuilder strBuild = new StringBuilder();

for (String s : wordArr) {

strBuild.append(s);

strBuild.append(" ");

}

str1 = strBuild.toString();

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

System.out.println(str1);

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

}

boolean containsAll(final Container 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;

}

@Override

public Iterator<String> iterator() {

return new IteratorMine();

}

public class IteratorMine implements Iterator<String> {

/\*\*

\* вложеный клас

\* унаследованый от интерфейса Iterator<String>

\*/

private int position = 0;

@Override

public boolean hasNext() {

return position < count;

}

@Override

public String next() {

if (this.hasNext()) {

return array[position++];

} else {

throw new NoSuchElementException();

}

}

@Override

public void remove() {

int temp = position;

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

array[temp++] = array[i + 1];

}

count--;

}

}

}

1. ВАРІАНТИ ВИКОРИСТАННЯ

Програму можна використовувати для зміни слова заданої довжини в тексті на слово, яке задає користувач.

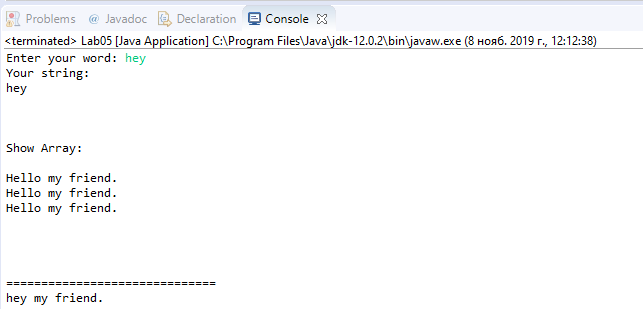


Рисунок 2 – Результат програми, масив речень та змінений текст

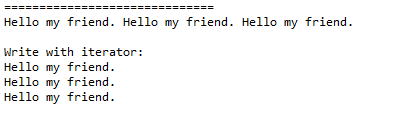


Рисунок 3 – Масив виведений у строку та виведення за допомогою ітераторів

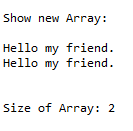


Рисунок 4 – Видалення елемента масиву та виведення розміру нового масиву

ВИСНОВОК

При виконанні лабораторної роботи набуто практичних навичок, щодо розробки власних утилітарних класів та задач з використанням масивів і рядків за допомогою JDK платформи Java SE, а також розробки власних контейнерів, з використанням ітераторів. Програма виконується без помилок.