Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Обчислювальна техніка та програмування»

ЗВІТ

Про виконання лабораторної роботи №5

“Розробка власних контейнерів. Ітератори”

Виконавець:

студент гр. КІТ-120В

Олексієнко Микита

Харків 2021

Лабораторна робота №5.

**Розробка власних контейнерів. Ітератори**

**Мета**

* Набуття навичок розробки власних контейнерів.
* Використання ітераторів.

**1.Вимоги**

**1.1 Розробник:**

Олексієнко Микита Віталійович

студент групи КІТ-120В

15 варіант

**1.2 Загальне завдання**

Реалізувати програму відповідно до індивідуального завдання.

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.
3. В класі ітератора відповідно до Interface Iterator реалізувати методи:
   * public boolean hasNext();
   * public String next();
   * public void remove().
4. Продемонструвати роботу ітератора за допомогою циклів while и for each.
5. Забороняється використання контейнерів (колекцій) і алгоритмів з Java Collections Framework.

**1.3 Прикладні задачи**

Ввести текст. У тексті знайти та вивести всі слова-паліндроми (однаково читається в обох напрямках - зліва направо та справа наліво. Наприклад: "noon", "civic", "radar", "level", "rotor", "refer").

**2. Опис програми**

**2.1 Засоби ООП:**

У лабороторного роботі 5 , я використовую клас Program, який э головним класом та в якому безпосередньо оголошую приватні поля і публічні статичні методи. Також я оголосив клас Palindromes, який використовую для обробки знайдених паліндромів та клас Menu. Він дозволяє використовувати введення та виведення даних.

Клас Container я використовую для зберігання введеного масиву рядків та їх обробки.

**2.2 Ієрархія та структура класів:**

* Program.java
* Palindomes.java
* Menu.java
* Container.java
* Iterator.java ( Interface )

**2.3 Важливі фрагменти програми:**

package ua.khpi.oop.alekseenko05;  
  
public class Container{  
 private int InputTextLength = 0;  
 private String[] InputText;  
  
 public void DEBUG () {  
 for ( String i : InputText ) {  
 System.out.println("-" + i + "-");  
 }  
 System.out.println();  
 }  
  
 public String textToString() {  
 StringBuilder result = new StringBuilder();  
  
 if ( InputText == null ) {  
 return "";  
 }  
  
 for ( String i : InputText ) {  
 result.append(i).append(" ");  
 }  
  
 return result.toString();  
 }  
  
 public void add(String text\_line) {  
 if ( InputTextLength == 0 ) {  
 InputTextLength++;  
 InputText = new String[InputTextLength];  
 InputText[0] = text\_line;  
 } else {  
 String[] temp = new String[InputTextLength];  
 temp = InputText;  
 InputTextLength++;  
 InputText = new String[InputTextLength];  
 for ( int i = 0; i < InputTextLength - 1; i++ ) {  
 InputText[i] = temp[i];  
 }  
 InputText[InputTextLength-1] = text\_line;  
 }  
 }  
  
 public void clear() {  
 InputText = new String[0];  
 InputTextLength = 0;  
 }  
  
 public boolean remove(String text\_line) {  
 for ( int i = 0; i < InputTextLength; i++ ) {  
 if ( InputText[i].equals(text\_line) ) {  
 // удаление со сдвигом+  
  
 String[] temp = new String[InputTextLength-1];  
  
 //arraycopy(1 массив, исходная позиция в 1 массиве, 2 массив, исходная позиция во 2 массиве, длина)  
  
 System.arraycopy(InputText, 0, temp, 0, i);  
 System.arraycopy(InputText, i + 1, temp, i, InputTextLength - i - 1);  
  
 InputText = temp;  
 InputTextLength--;  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public Object[] toArray() {  
 Object[] objects = new Object[2];  
 objects[0] = InputText;  
 objects[1] = InputTextLength;  
 return objects;  
 }  
  
 public boolean contains(String text\_line) {  
 for ( int i = 0; i < InputTextLength; i++ ) {  
 if ( InputText[i].equals(text\_line) ) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public boolean containsAll(Container container) {  
 for ( int i = 0; i < InputTextLength; i++ ) {  
 if ( container.InputText == InputText  
 && container.InputTextLength == InputTextLength ) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public int getSize () {  
 return InputTextLength;  
 }  
 public void setSize ( int number ) { InputTextLength = number; }  
  
 public Iterator getIterator() {  
 return new ArrayIterator();  
 };  
  
 public class ArrayIterator implements Iterator {  
 int index;  
  
 @Override  
 public boolean hasNext() {  
 return index < InputTextLength;  
 }  
  
 public String next() {  
 if ( hasNext() ) {  
 return InputText[index++];  
 }  
 return null;  
 }  
  
 public void remove() {  
 for ( int i = 0; i < InputTextLength; i++ ) {  
 if ( i == index ) {  
 String[] temp = new String[InputTextLength-1];  
  
 //arraycopy(1 массив, исходная позиция в 1 массиве, 2 массив, исходная позиция во 2 массиве, длина)  
  
 System.arraycopy(InputText, 0, temp, 0, i);  
 System.arraycopy(InputText, i + 1, temp, i, InputTextLength - i - 1);  
  
 InputText = temp;  
 InputTextLength--;  
 }  
 }  
 }  
  
 public void reset() {  
 index = 0;  
 }  
 }  
}

**3. Висновок:**

* Набув навичок розробки власних контейнерів.
* Навчився створювати та використовувати ітератори.