Лабораторна робота №5

Розробка власних контейнерів. Ітератори

Мета:набуття навичок розробки власних контейнерів. Використання ітераторів.

1 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(Stringstring) додає вказаний елемент до кінця контейнеру;
    - Void clear() видаляє всі елементи з контейнеру;
    - booleanremove(Stringstring) 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(Stringstring) повертає true, якщо контейнер містить вказаний елемент;
    - Boolean containsAll(Containercontainer) 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 InterfaceIterator реалізувати методи:
    - Public boolean hasNext();
    - Public String next();
    - Public void remove().
4. Продемонструвати роботу ітератора за допомогою циклів while и foreach.
5. Забороняється використання контейнерів (колекцій) і алгоритмів з JavaCollections Framework.

1.2 Розробник: Мітін Микита Валерійович КІТ119д №15.

1.2 Задача

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

2.1Було використано наступні засоби:

- StringBuilder = newStringBuilder()–створення StringBuilder;
- Iterator<String> it = container.getIterator() – Ітератор;

2.2 Ієрархія та структура класів

Було створено 3 класи:

- Public class Container – клас, що реалізує методи контейнеру.
- Public class My_iterator – клас, що реалізує методи ітератора.
- public class Main – містить лише метод main.

Важливі фрагменти програми

Клас Container

```
package ua.khpi.oop.mitin05;

import java.util.Iterator;

public class Container {
	private String [] container;
	private int size;

	public String toString() //возвращает содержимое контейнера в виде строки;
	{

		String str = "";
		for (String string : container) {
			str += string + " ";
		}
		return str;

	}

	public void add(String str) //добавление элемента в конец контейнера
	{
		int size = container.length;
		String [] new_container = new String[size+1];
		for (int i=0;i<size;i++) {
			new_container[i]=container[i];
		}
		new_container[size]=str;
```

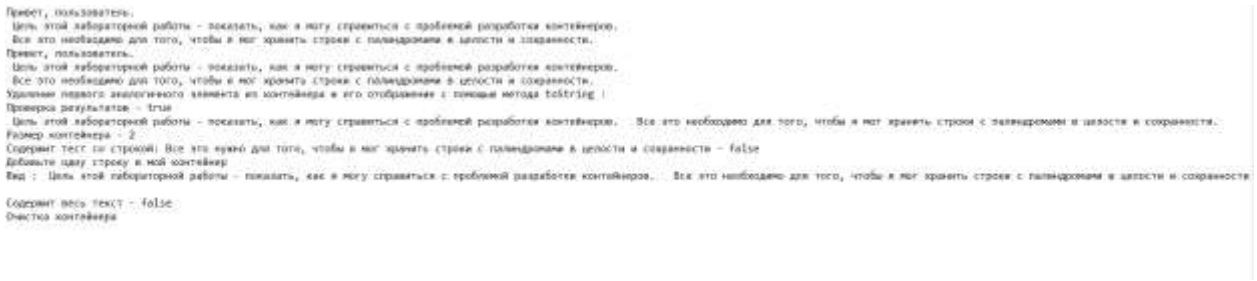
```
        size++;
        container = new_container;
    }

    public void clear() //удаляет все элементы контейнера
    {
        for (int i = 0; i < container.length; i++) {
            container[i]=null;

        }
        size =0;
    }

    public boolean remove(String str) // удаление элемента
    {

        boolean flag = false;
        String [] new_container = new String[size-1];
        for(int i=0;i<size;i++) {
            if(container[i].equals(str))
                flag = true;
        }
        if(flag) {
            for(int i=0,j=0;i<size;i++) {
                if(container[i].equals(str))
                    i++;
                new_container[j]=container[i];
                j++;
            }
            size--;
            container = new_container;
```

```
                return flag;
            }
            else
            {
                return flag;
            }
        }

        public String[] toArray()  //возвращает массив, содержащий все элементы контейнера
        {
            return container;
        }

        public int size()  // возвращает кол-во элементов
        {
            return size;
        }

        public boolean containsAll(Container cont) //возвращает true, если контейнер содержит все элементи из указанного в параметрах;
        {
            int count = 0;
            for (int i = 0; i < container.length; i++) {
                for (int j = 0; j < cont.container.length; j++) {
                    if(cont.container[j].equals(container[i]))
                        count++;
                }
            }
            if(count == cont.container.length)
                return true;
            else
```

```
                return false;
        }



        public boolean contains(String str) //возвращает true, если контейнер содержит указанный элемент;
        {
                boolean flag = false;
                for (String string : container) {
                        if(string.equals(str))
                                flag=true;
                }
                return flag;
        }




        public Container(String... str) {
                if(str.length!=0) {
                        size = str.length;
                        container = new String[size];
                        for (int i=0;i<size;i++) {
                                container[i]=str[i];
                        }
                }
        }



        public Iterator<String> getIterator() {
                return new My_iterator<String>();
```

```
	}


	public class My_iterator<String> implements Iterator {
		int index;


		@Override
		public boolean hasNext() {
			return index < size ? true : false;
		}


		@Override
		public Object next() {
			return container[index++];
		}


		/*Method that removes from the underlying collection the last element
returned by this iterator*/
		@Override
		public void remove() {
			Container.this.remove(container[--index]);
		}
	}
}
```

Результат роботи програми:



ВИСНОВКИ

У результаті виконання лабораторної роботи було набуто навичок з розробки власних контейнерів, роботи з ітераторами у середовищі JavaEclipse.