Лабораторна робота №5

РОЗРОБКА ВЛАСНИХ КОНТЕЙНЕРІВ. ІТЕРАТОРИ

Мета: набуття навичок розробки власних контейнерів, використання ітераторів.

Основне 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.

ЗМІСТ

2.ОПИС ПРОГРАМИ

2.1. Розробка програми

Для виконання лабораторної роботи було створено нові класи для контейнеру та спеційного ітератору. Також було використано утилітарний клас з попередньої лабораторної роботи. Програма виконує завдання попередньої лабораторної, тобто виводить на екран найдовші та найкоротші слова.

2.2. Важливі фрагменти програми

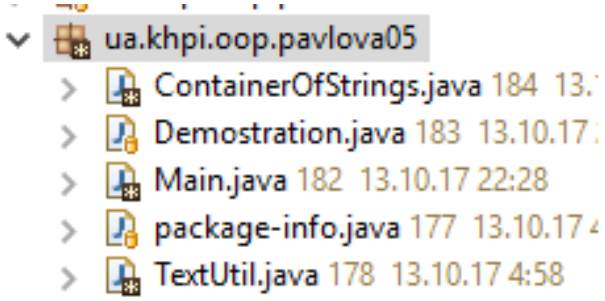

рис.1. Пакет лабораторної роботи

```java
/**
 * Method <b>toString</b> converts an object of <b>ContainerOfStrings</b> to a
 * string.
 */
public String toString() {
    String containerContent = new String();
    for (String string : elementData)
        containerContent += string;
    return containerContent;
}
```

рис.2. toString()

```java
/**
 * Method <b>add</b> writes a new one string to the container. The capacity
 * increases while adding a new string.
 *
 * @param string
 *            is a new one line in a container
 */
public void add(String string) {
    if (size == elementData.length) {
        ensureCapacity(size + 1);
    }
    elementData[size] = string;
    size++;
}
```

рис.3. add()

```java
/**
 * Method <b>clear</b> performs the removal of all the elements in the
 * container. Capacity is saved, but the size decreases to null value.
 */
public void clear() {
    for (int i = 0; i < size; i++)
        elementData[i] = null;

    size = 0;
}
```

рис.4. clear()

```java
/**
 * Method <b>remove</b> performs the removal of the concreate string in the
 * container. If this string is absent the false value returns.
 *
 * @param string
 *            is a string that has to be removed from the container
 * @return true/false value
 */
public boolean remove(String string) {
    for (int i = 0; i < elementData.length; i++) {
        if (string.equals(elementData[i])) {
            int removed = elementData.length - i - 1;
            if (removed > 0) {
                System.arraycopy(elementData, i + 1, elementData, i, removed);
            }
            elementData[size - 1] = null;
            return true;
        }
    }
    return false;
}
```

рис.5. remove()

```java
/**
 * Method <b>toArray</b> creates an array of <b>Object</b> type from the
 * container.
 *
 * @return an array of objects
 */
public Object[] toArray() {
    return copy();
}
```

рис.6. toArray()

```java
/**
 * Method <b>copy</b> is an accessory one. Creates a copy of an array of strings
 * in the object format.
 *
 * @return a copy of an array
 */
public Object[] copy() {
    Object[] copy = (String[]) new Object[size];
    System.arraycopy(elementData, 0, copy, 0, Math.min(elementData.length, size));
    return copy;
}
```

рис.7. copy()

```java
/**
 * Method <b>size</b> gets the value of container's size
 *
 * @return size of the container
 */
public int size() {
    return size;
}
```

рис.8. size()

```java
/**
 * Method <b>contains</b> performs the check-up of availability of a string in
 * the container. If it exists - returns true, else - false.
 *
 * @param string
 *            a line for the check-up
 * @return true/false value
 */
public boolean contains(String string) {
    int temp = 0;
    if (string == null) {
        for (int i = 0; i < size; i++)
            if (elementData[i] == null)
                temp = i;
    } else {
        for (int i = 0; i < size; i++)
            if (string.equals(elementData[i]))
                temp = i;
    }
    temp = -1;
    return temp >= 0;
}
```

рис.9. contains()

```java
/**
 * Method <b>iterator</b> returns a new iterator for the container
 */
public IteratorForContainer<String> iterator() {
    return new IteratorForContainer<String>(elementData);
}
```

рис.10. Iterator()

Реалізація ітератору:

```java
@Override
public boolean hasNext() {
    return this.cursor < end;
}
```

рис.11. HasNext()

```java
@SuppressWarnings("unchecked")
@Override
public String next() {
    if (!this.hasNext()) {
        throw new NoSuchElementException();
    }
    cursor++;
    return (String) elementData[cursor];
}
```

рис.12. Next()

```
public void remove() {
    if (end == -1)
        throw new IllegalStateException();
    try {
        ContainerOfStrings.this.remove(elementData[end]);
        if (end < cursor)
            cursor--;
        end = -1;
    } catch (IndexOutOfBoundsException one) {
        throw new ConcurrentModificationException();
    }
}
```

рис.13. Remove()

```
System.out.println("Список найдовших слів");
ContainerOfStrings.IteratorForContainer<String> iteratorForContainerLongest = longest.iterator();
while (iteratorForContainerLongest.hasNext()) {
    System.out.println(iteratorForContainerLongest.next());
}
```

рис.14. Цикл while

```
for (String shortWord : shortest) {
    System.out.println(shortWord);
}
```

рис.15. Цикл for each

3. РЕЗУЛЬТАТИ

```
Лабораторна робота №5
Номер прикладної задачі: 3
Bhfs fjch rifi vjv eon dj dj ejejcj. Jnkjwn wjn cjw wur pirjwojf wfjn wrfn w. Jnv wivjw jv wjv wvj wi. Jwr vjkwns vsfnk.
end
Список найдовших слів
ejejcj
pirjwojf
wivjw
vjkwns
Список найкоротших слів
dj
w
jv
Jwr
```

рис.16. Результати

4. ВИСНОВКИ

В результаті виконання лабораторної роботи ми навчились писати власні контейнери та працювати із ітераторами.