

5. Розробка власних контейнерів. Ітератори

Мета: Набуття навичок розробки власних ітераторів. Використання ітераторів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кононенко Дмитро Олексійович
- НТУ “ХПІ” 1.KIT102.8a
- Варіант 8

1.2 Загальне 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*.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі відсутні об'єктно-орієнтовані методи.

2.2 Ієрархія та структура даних

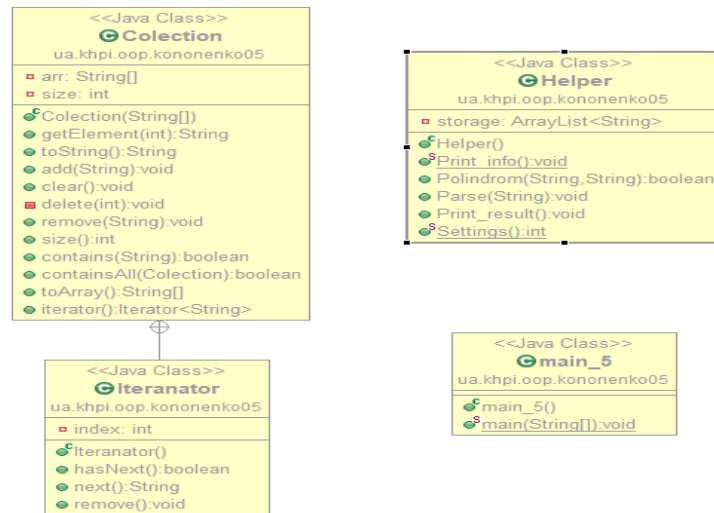


Рисунок 1 – діаграма класів

2.3 Важливі фрагменти програми

```
123 public class Iteranator implements Iterator<String>{
124     private int index = 0;
125
126     @Override
127     public boolean hasNext() {
128         if(index < size && arr[index] != null) {
129             return true;
130         }
131         return false;
132     }
133
134     @Override
135     public String next() {
136         return arr[index++];
137     }
138
139     @Override
140     public void remove() {
141         throw new UnsupportedOperationException();
142     }
143 }
144
145 public Iterator<String> iterator() {
146     return new Iteranator();
147 }
148 }
149
```

Рисунок 2 – тіло класу ArrayIterator

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма створена задля демонстрації використання та тестування різноманітних методів розробленого контейнера, що містить масив рядків.

```
To string:
Hello it is me mario and oiram It is gonna be hot here si We should find better place Oh hello there olleh
To array:
Hello it is me mario and oiram
It is gonna be hot here si
We should find better place
Oh hello there olleh
It is gonna be hot here si
-----
|   Word   |   Reverse word   |
-----
|    is    |    si    |
|    si    |    is    |
```

ВИСНОВКИ

В даній лабораторній роботі розробив та використав власний ітератор задля зберігання і обробки елементів контейнера.