

Пример за списъци

Премахване на елементи от ArrayList с
помощтта на Iterator

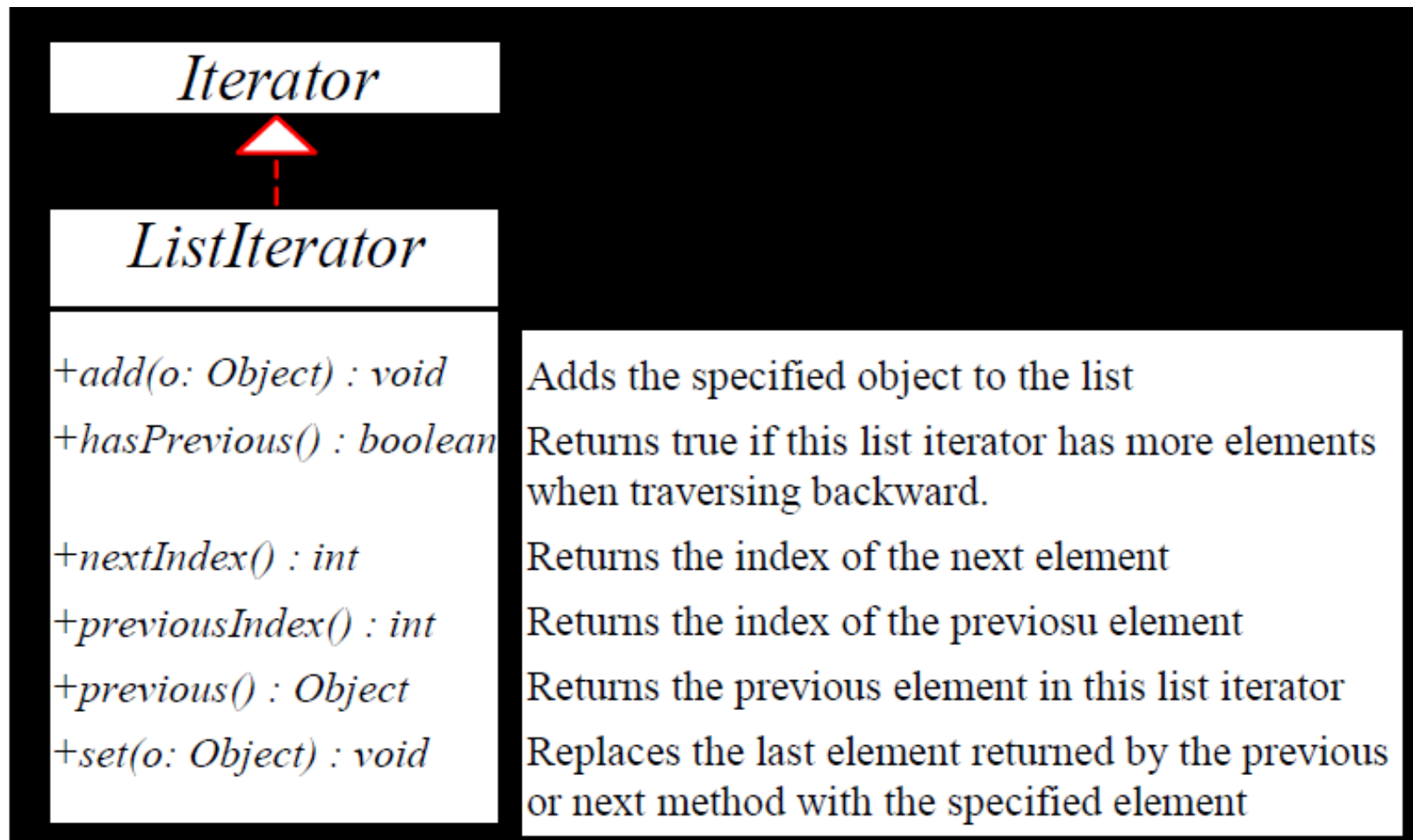
ArrayList и Iterator

Обхождане на списък с помощта на итератор.

*Програмата въвежда два масива с имена на цветове в **ArrayList** колекции и използва **Iterator** да изтрие елементите от първата **ArrayList** колекция, които са дублират елементи във втората **ArrayList** колекция*

- Демонстрира методи на **interface Collection**
- Въвежда **String** масиви в **ArrayList** колекции
- Използва **Iterator** за изтриване на елементи от **ArrayList** колекция

ArrayList и Iterator



```
1 // CollectionTest.java
2 // Using the Collection interface.
3 import java.util.List;
4 import java.util.ArrayList;
5 import java.util.Collection;
6 import java.util.Iterator;
7
8 public class CollectionTest
9 {
10     private static final String[] colors =
11         { "MAGENTA", "RED", "WHITE", "BLUE", "CYAN" };
12     private static final String[] removeColors =
13         { "RED", "WHITE", "BLUE" };
14
15     // create ArrayList, add Colors to it and manipulate it
16     public CollectionTest()
17     {
18         List< String > list = new ArrayList< String >();
19         List< String > removeList = new ArrayList< String >();
20     }
```

Създава **ArrayList** обекти и присвоява референции към тях съответно на променливи **list** и **removeList**

```
21 // add elements in colors array to list
22 for ( String color : colors )
23     list.add( color );
24
25 // add elements in removeColors to removeList
26 for ( String color : removeColors )
27     removeList.add( color );
28
29 System.out.println( "ArrayList: " );
30
31 // output list contents
32 for ( int count = 0; count < list.size(); count++ )
33     System.out.printf( "%s ", list.get( count ) );
34
35 // remove colors contained in removeList
36 removeColors( list, removeList );
37
38 System.out.println( "\n\nArrayList after calling removeColors: " );
39
40 // output list contents
41 for ( String color : list )
42     System.out.printf( "%s ", color );
43 } // end CollectionTest constructor
44
```

Исползва метод **add** на **List** за добавяне на елемент към **list** и **removeList**, съответно

Исползва метод **size** на **List** за получаване на елементите на **ArrayList**

Исползва метод **get** на **List** за четене на отделни елементи

Метод **removeColors** приема два аргумента от тип **Collection**; Ред 36 предава като аргументи два **List** обекта, които се преобразуват до **Collection**

```

45 // remove colors specified in collection2 from collection1
46 private void removeColors(
47     Collection< String > collection1, Collection< String > collection2 )
48 {
49     // get iterator
50     Iterator< String > iterator = collection1.iterator();
51
52     // loop while collection has items
53     while ( iterator.hasNext() )
54     {
55         if ( collection2.contains( iterator.next() ) )
56             iterator.remove(); // remove current Color
57     } // end method removeColors
58
59     public static void main( String args[] )
60     {
61         new CollectionTest();
62     } // end main
63 } // end class CollectionTest

```

Метод **removeColors** работи с **Collection** от **String** аргументи

Получава **итератор** за **Collection** обект

Iterator метода **hasNext** определя дали **Iterator** има още **елементи** за обхождане

Iterator метода **next** връща **референция към следващия** елемент в колекцията

ArrayList:
MAGENTA RED WHITE BLUE CYAN

ArrayList after calling removeColors:
MAGENTA CYAN

Използва метода **remove** на **Iterator** за **изтриване** на **String** от **collection1**

Методът **contains** на **Collection** определя дали **collection2** съдържа елемента върнат с **next**