GEBZE TECHNICAL UNIVERSITY

COMPUTER ENGİNEERİNG

DATA STRUCTURES AND ALGORITHMS

HOMEWORK #3

REPORT

MUHAMMED YASİR FİDAN

161044056

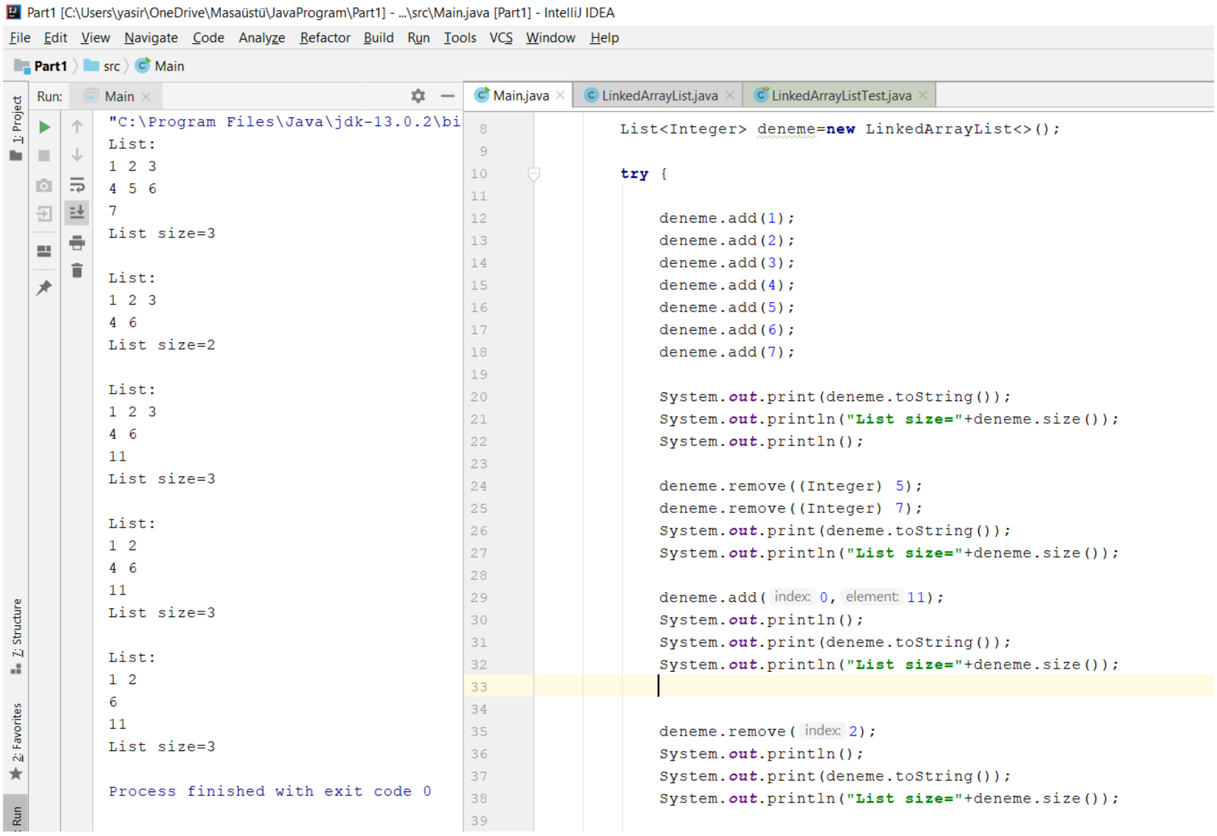
PART1

My LinkedArrayList class extends AbstractList and implements List interface and My LinkedArrayList class has 2 inner classes Node and MyIterator class. I declare Node class as static because Node class doesnt need to Access LinkedArrayList fields or methods. This class hold an array with its size and capacity.In my test program array capacity is 3 . You can change it.Node class also has 2 references for points the next node and previous node.My LinkedArrayList is a double linked list so I have a head referance and tail referance. There is also size field for linklist. myIterator is my other inner class for iterate the LinkedArrayList. I implement myIterator class to the ListIterator Interface and override its some methods for use correctly in my new data structure as a Iterator.

My LinkedArrayList can add some elements to the Nodes arrays and can remove some element to those arrays.There is 2 add method in my class.First add only take one paramater. This add method Works like a ArrayList add method. It goes to last node and add the argument to the last array element, but if array capacity is full it create a new Node at the end and add this element to this node’s array.If you want to create new node first you must fill a node’s array then automatically new nodes will be created at the end of the list.The other add method takes an element for adding and an index. But be carefull index must be bigger than 0 and smaller than node size, if it is not you will get an IndexOutOfBoun excaption. This method goes the correspoinding index node and add element to its array.But if this array capacity is full then like the first add method it creates a new node at the end of the list and add this element to this node.

There is 2 remove method the first one take an element and traverse list node by node to find the array that has this element. When its find an array that has this element its remove this element from the array and shift other elements.If the size of array became 0 after the remove method then it will also remove the node from the LinkedArrayList structure.

The other remove method take an index as paramater and remove the element where in that index.


Part1 [C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\Part1] - ...\src\Main.java [Part1] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Part1 〉 src 〉 Main
Run: Main
Main.java
LinkedArrayList.java
LinkedArrayListTest.java
1: Project
7: Structure
2: Favorites
: Run

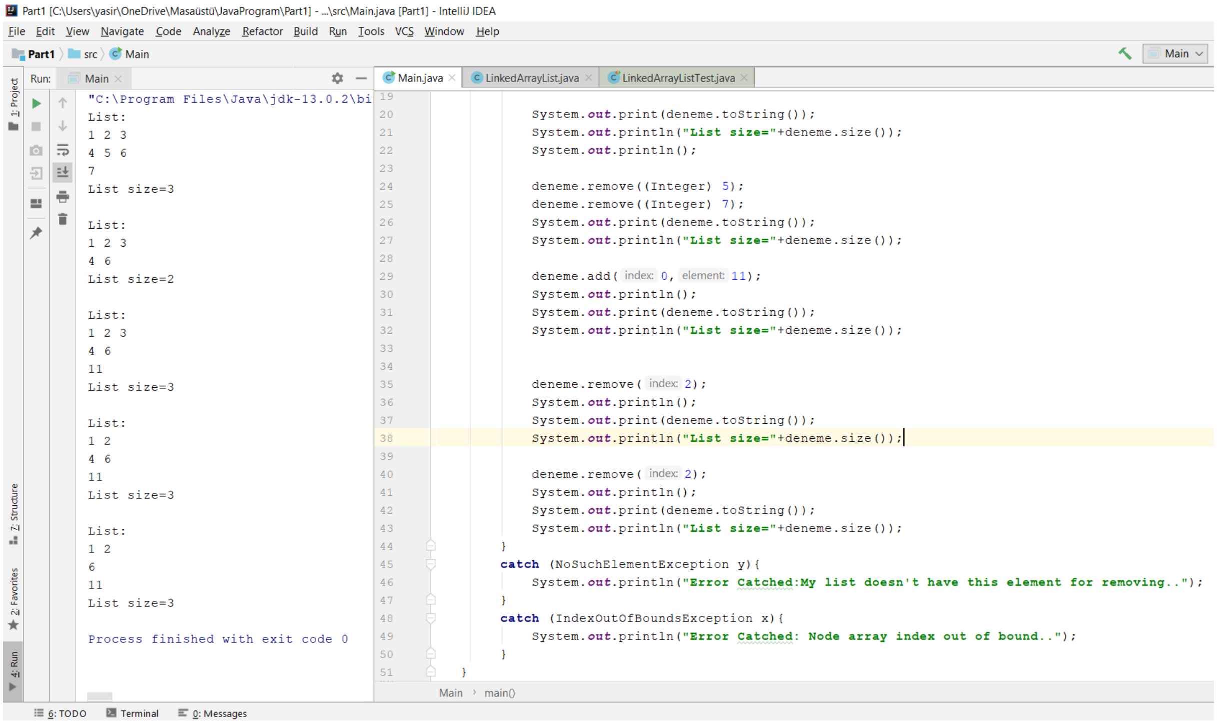
```
"C:\Program Files\Java\jdk-13.0.2\bi
List:
1 2 3
4 5 6
7
List size=3

List:
1 2 3
4 6
List size=2

List:
1 2 3
4 6
11
List size=3

List:
1 2
4 6
11
List size=3

List:
1 2
6
11
List size=3

Process finished with exit code 0
```

```
8           List<Integer> deneme=new LinkedArrayList<>();
9   
10          try {
11  
12              deneme.add(1);
13              deneme.add(2);
14              deneme.add(3);
15              deneme.add(4);
16              deneme.add(5);
17              deneme.add(6);
18              deneme.add(7);
19  
20              System.out.print(deneme.toString());
21              System.out.println("List size="+deneme.size());
22              System.out.println();
23  
24              deneme.remove((Integer) 5);
25              deneme.remove((Integer) 7);
26              System.out.print(deneme.toString());
27              System.out.println("List size="+deneme.size());
28  
29              deneme.add( index: 0, element: 11);
30              System.out.println();
31              System.out.print(deneme.toString());
32              System.out.println("List size="+deneme.size());
33              
34  
35              deneme.remove( index: 2);
36              System.out.println();
37              System.out.print(deneme.toString());
38              System.out.println("List size="+deneme.size());
39  
```

In this test program I created a linkedArrayList object and try to methods to Show they work correctly.In this example my arrays capacity is 3. In line 12 first add(1) also create a node because there is no node at the beginning after that it add 1 to this node array first index.After line 13 and 14 I add 2 and 3 this elements added to the first node array second and third index.In line 15 when I try to add 4 it creates a Node at the end of the list because our last Node array was full(it contains 1 2 3 so its full) and its add 4 to this new node array.Similary When I call add(5) and add(6) its go to the last node and add elemets to this nodes array but when I try to add 7 because of my array was full(it contains 4 5 6 so its full) it creates a new node and add element 7 to this nodes array.After that I call print method top rint list.Note that in output screen every line is a node and the elemets of the line is the elements of the array that belong to the node.As you can see there is 3 node so when I try to print the size of the LinkedArrayList it prints 3.

After that in line 24 I call remove(5). This remove method traverse the list with my Iterator(myIterator) and find the element 5 then remove it and shift the array. as you can see (4 5 6) became (4 6). Line 25 call remove(7) so its traverse the list and find element 7 and remove it but after the remove the array size became 0 so this remove also remove this node too so my list size became 2.

Lastly, in line 29 I call other add method that takes an element and index as argument.Here index 0 and element 11 so this add try to add element 11 at first Node but the first nodes array is full so it create a new node at the end of the list and add element 11 into this nodes array.

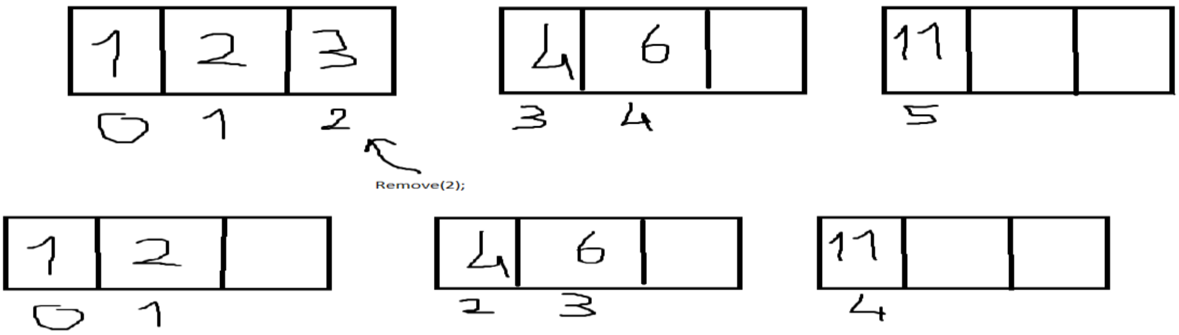

Part1 [C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\Part1] - ...\src\Main.java [Part1] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Part1 src Main
Main
Run: Main
Main.java
LinkedArrayList.java
LinkedArrayListTest.java
1: Project
7: Structure
2: Favorites
4: Run
Main › main()
6: TODO
Terminal
0: Messages

```
"C:\Program Files\Java\jdk-13.0.2\bi
List:
1 2 3
4 5 6
7
List size=3

List:
1 2 3
4 6
List size=2

List:
1 2 3
4 6
11
List size=3

List:
1 2
4 6
11
List size=3

List:
1 2
6
11
List size=3

Process finished with exit code 0
```

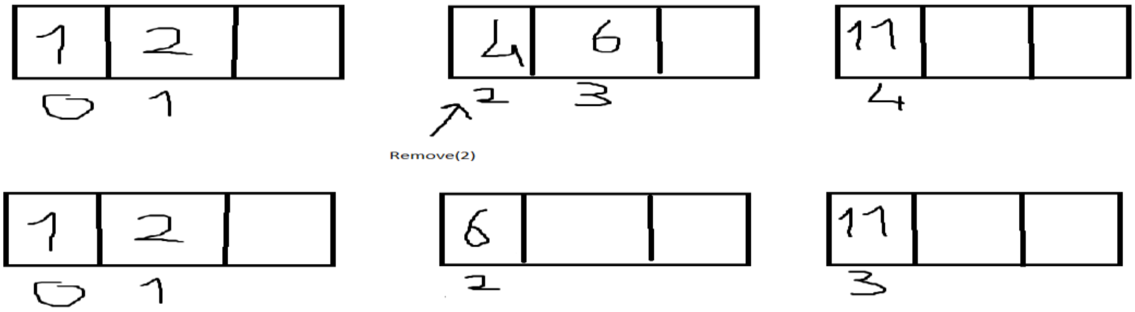
```
19
20              System.out.print(deneme.toString());
21              System.out.println("List size="+deneme.size());
22              System.out.println();
23
24              deneme.remove((Integer) 5);
25              deneme.remove((Integer) 7);
26              System.out.print(deneme.toString());
27              System.out.println("List size="+deneme.size());
28
29              deneme.add( index: 0, element: 11);
30              System.out.println();
31              System.out.print(deneme.toString());
32              System.out.println("List size="+deneme.size());
33
34
35              deneme.remove( index: 2);
36              System.out.println();
37              System.out.print(deneme.toString());
38              System.out.println("List size="+deneme.size());
39
40              deneme.remove( index: 2);
41              System.out.println();
42              System.out.print(deneme.toString());
43              System.out.println("List size="+deneme.size());
44          }
45          catch (NoSuchElementException y){
46              System.out.println("Error Catched:My list doesn't have this element for removing..");
47          }
48          catch (IndexOutOfBoundsException x){
49              System.out.println("Error Catched: Node array index out of bound..");
50          }
51      }
```

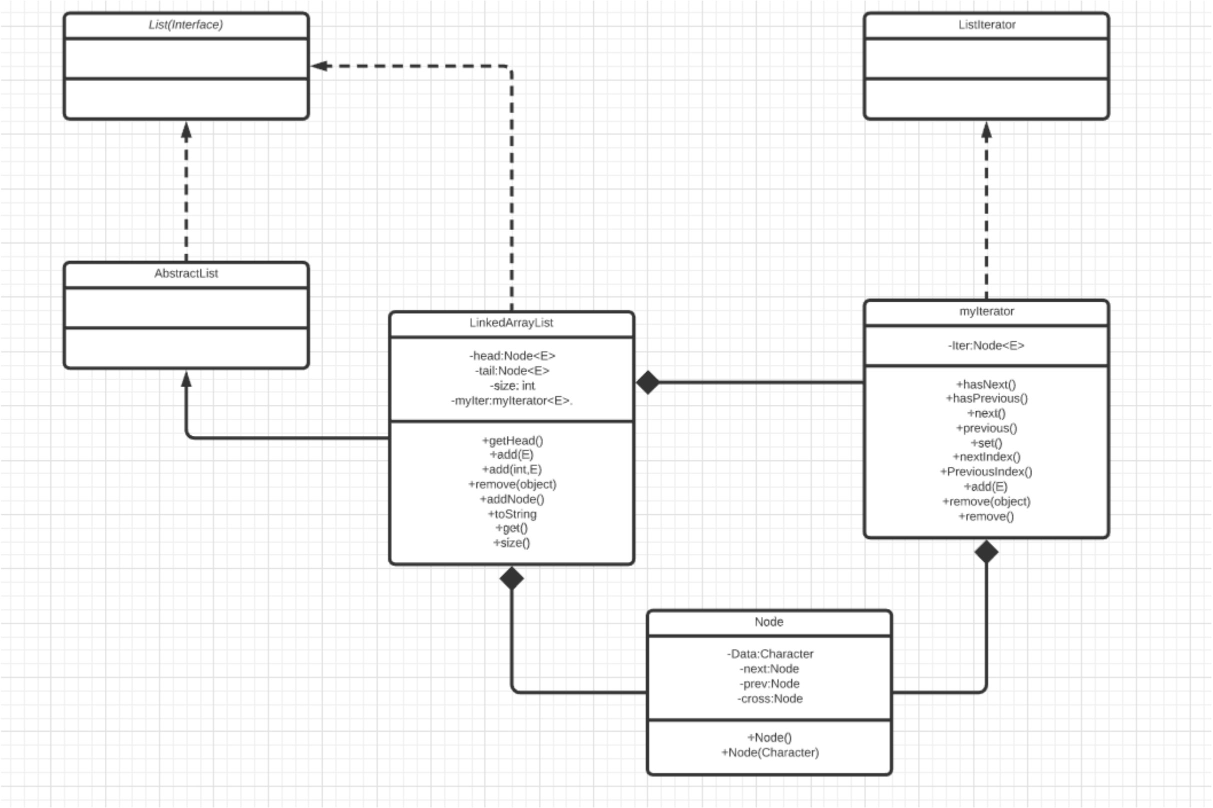
In line 35 I call other remove method that take index as paramater.Index =2 so it remove 3 than print the list after that again I call remove index method with index 2 and it remove 4 and shift 6.

After First Remove(2):



After second Remove(2):

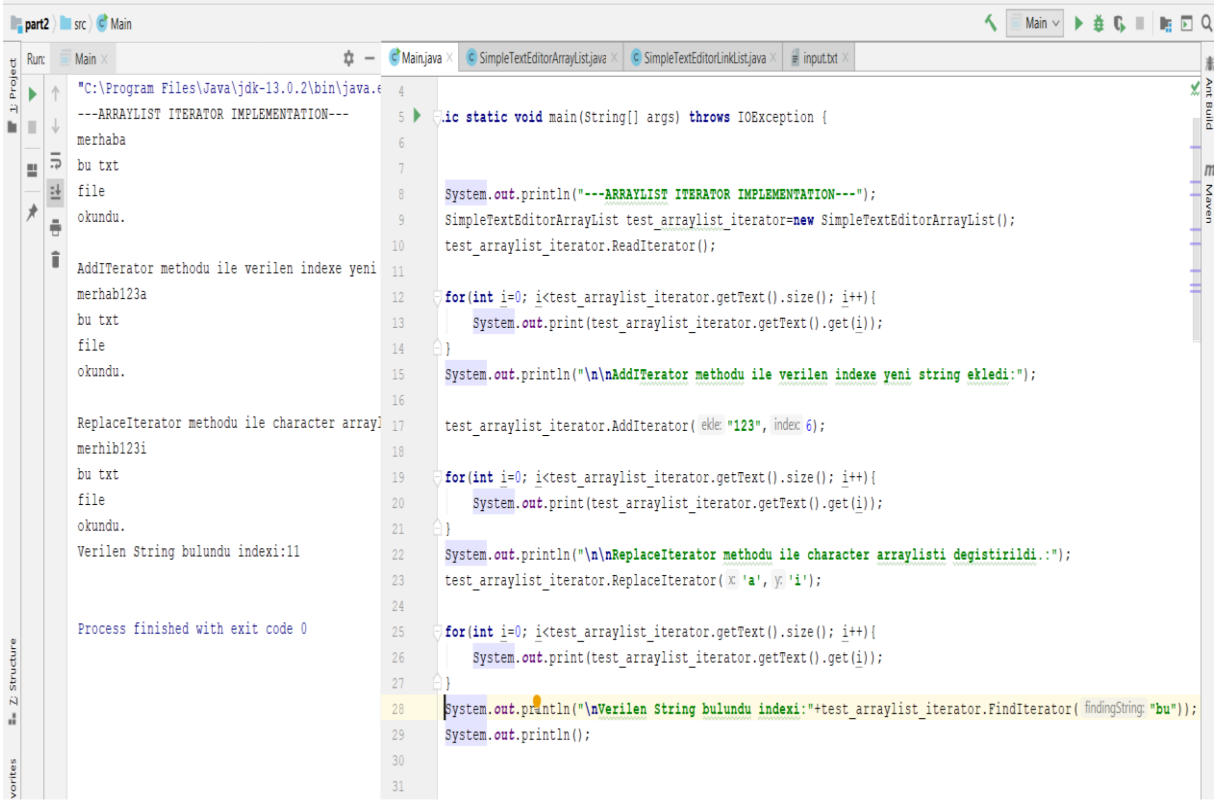




Part1 class Diagram

PART2

In part2, I implemen SimpleTextEditorArrayList and SimpleTextEditorLinkList classes but these two class is same only difirence is SimpleTextEditorArrayList use a array list but SimpleTextEditorLinkList use a linkedlist.Each class has 8 methods.4 method implement with iterator(ReadIterator,ReplaceIterator,AddIterator,FindIterator) and the other 4 method implement with simple loop(Read,Replace,Add,Find).In my main I create 4 object and do same program with this objects.These object for test_arraylist_iterator, test_arraylist_loop,test_linklist_iterator and test_linklist_loop. So These object call corresponding 4 method.



```
"C:\Program Files\Java\jdk-13.0.2\bin\java.e
---ARRAYLIST ITERATOR IMPLEMENTATION---
merhaba
bu txt
file
okundu.

AddITerator methodu ile verilen indexe yeni
merhab123a
bu txt
file
okundu.

ReplaceIterator methodu ile character arrayl
merhib123i
bu txt
file
okundu.
Verilen String bulundu indexi:11


Process finished with exit code 0
```

```
4
5   .ic static void main(String[] args) throws IOException {
6
7
8   System.out.println("---ARRAYLIST ITERATOR IMPLEMENTATION---");
9   SimpleTextEditorArrayList test_arraylist_iterator=new SimpleTextEditorArrayList();
10  test_arraylist_iterator.ReadIterator();
11
12  for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
13      System.out.print(test_arraylist_iterator.getText().get(i));
14  }
15  System.out.println("\n\nAddITerator methodu ile verilen indexe yeni string ekledi:");
16
17  test_arraylist_iterator.AddIterator( ekle: "123", index: 6);
18
19  for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
20      System.out.print(test_arraylist_iterator.getText().get(i));
21  }
22  System.out.println("\n\nReplaceIterator methodu ile character arraylisti degistirildi.:");
23  test_arraylist_iterator.ReplaceIterator( x: 'a', y: 'i');
24
25  for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
26      System.out.print(test_arraylist_iterator.getText().get(i));
27  }
28  System.out.println("\nVerilen String bulundu indexi:"+test_arraylist_iterator.FindIterator( findingString: "bu"));
29  System.out.println();
30
31
```

Here I Show only Arraylist Iterator object but other objects do same thing.In line 10 I read input.txt file(merhaba\nbu txt\nfile\nokundu.).in line 17 I call AddIterator method with “123” paramater and index 6. So its add 123 in the 6 index of the list.so merhaba became merhab123a.In line 23 I call ReplaceIterator method with paramaters a and i so in my list its replace every ‘a’ with ‘i’.In line 28 I call FindIterator method with “bu” parameter and its return index 11 because first “bu” in index 11.

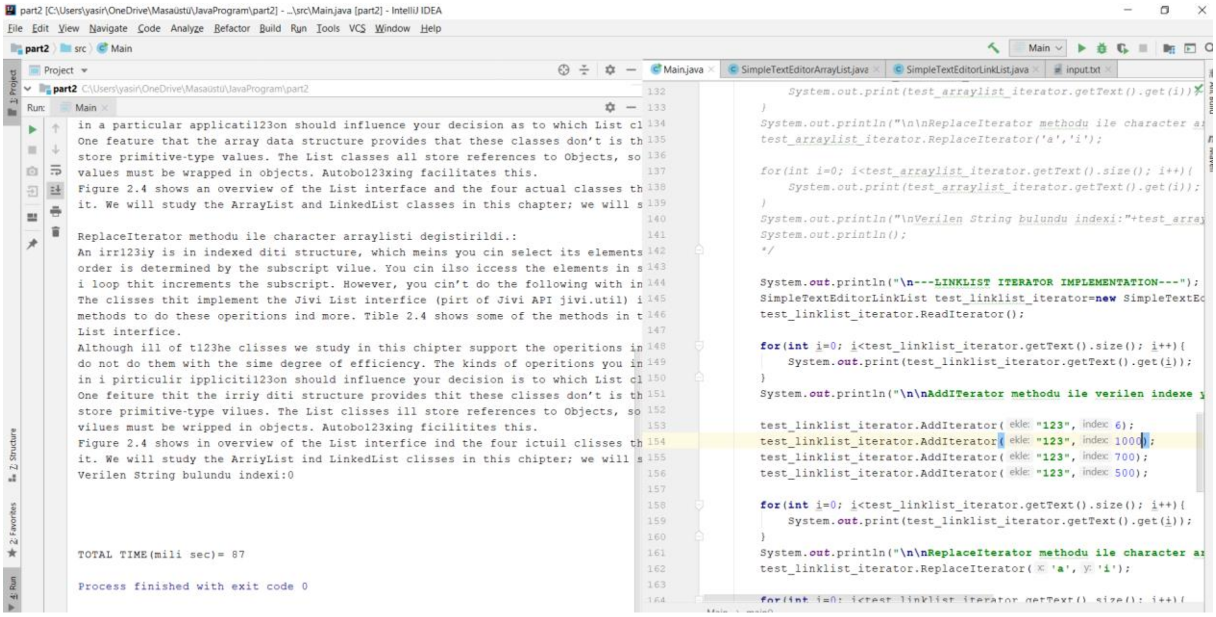
asymptotic notation

- **List is an ArrayList and iterator is used**
    - **ReadIterator() → O(n^2) because of Reallocation of arraylist**
    - **AddIterator() → O(n^2) because of Reallocation of arraylist**
    - **FindIterator() → O(n^2)**
    - **ReplaceIterator() →O(n)**
- **List is an ArrayList and iterator is not used**
    - **Read() → O(n^2) Because of Reallocation of arraylist**
    - **Add() → O(n^2) because of Reallocation of arraylist**
    - **Find() → O(n^2)**
    - **Replace() → O(n)**
- **List is a LinkedList and iterator is used**
    - **ReadIterator() →O(n)**
    - **AddIterator() →O(n)**
    - **FindIterator() →O(n^2)**
    - **ReplaceIterator()→O(n)**
- **List is a LinkedList and iterator is not used**

- Read() →O(n^2)
- Add() →O(n^2)
- Find() →O(n^3)
- Replace() →O(n^3)

Experimental results:

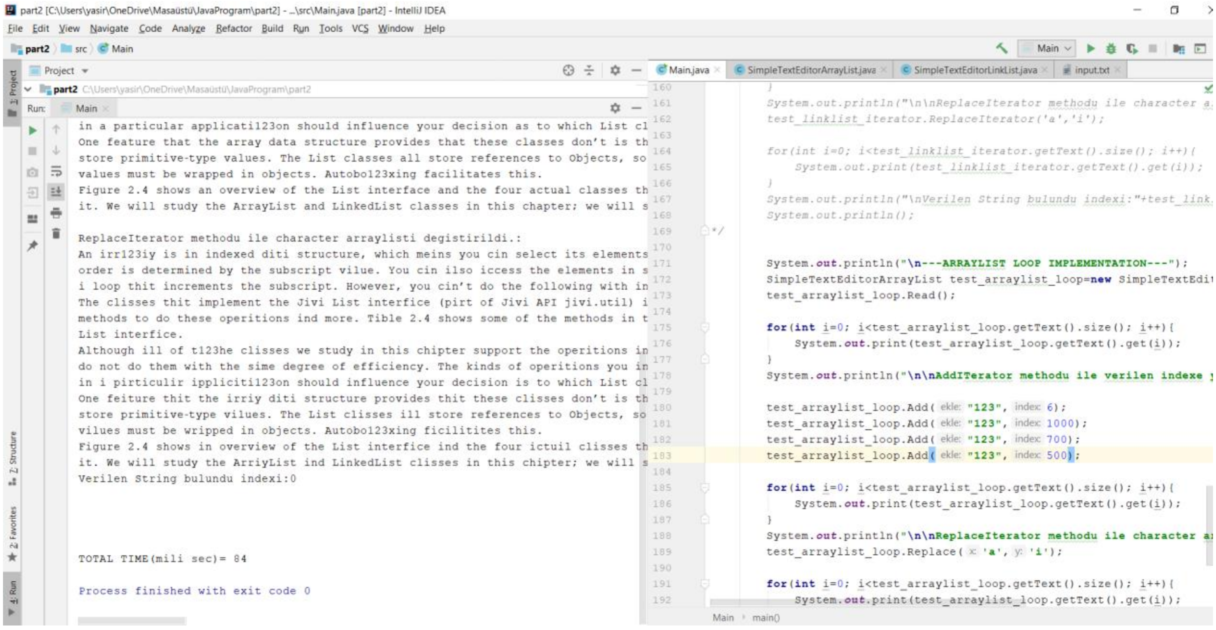
LinkList ITerator


part2 [C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\part2] - ...\src\Main.java [part2] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part2 src Main
Project
part2 C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\part2
Run: Main
Main.java SimpleTextEditorArrayList.java SimpleTextEditorLinkList.java input.txt
Main
1: Project
Z: Structure
2: Favorites
4: Run
Main › main()

```
in a particular applicati123on should influence your decision as to which List cl
One feature that the array data structure provides that these classes don't is th
store primitive-type values. The List classes all store references to Objects, so
values must be wrapped in objects. Autobo123xing facilitates this.
Figure 2.4 shows an overview of the List interface and the four actual classes th
it. We will study the ArrayList and LinkedList classes in this chapter; we will s

ReplaceIterator methodu ile character arraylisti degistirildi.:
An irr123iy is in indexed diti structure, which meins you cin select its elements
order is determined by the subscript vilue. You cin ilso iccess the elements in s
i loop thit increments the subscript. However, you cin't do the following with in
The clisses thit implement the Jivi List interfice (pirt of Jivi API jivi.util) i
methods to do these operitions ind more. Tible 2.4 shows some of the methods in t
List interfice.
Although ill of t123he clisses we study in this chipter support the operitions in
do not do them with the sime degree of efficiency. The kinds of operitions you in
in i pirticulir ippliciti123on should influence your decision is to which List cl
One feiture thit the irriy diti structure provides thit these clisses don't is th
store primitive-type vilues. The List clisses ill store references to Objects, so
vilues must be wripped in objects. Autobo123xing ficilitites this.
Figure 2.4 shows in overview of the List interfice ind the four ictuil clisses th
it. We will study the ArriyList ind LinkedList clisses in this chipter; we will s
Verilen String bulundu indexi:0




TOTAL TIME(mili sec)= 87

Process finished with exit code 0
```

```java
        System.out.print(test_arraylist_iterator.getText().get(i));
    }
    System.out.println("\n\nReplaceIterator methodu ile character a
    test_arraylist_iterator.ReplaceIterator('a','i');

    for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
        System.out.print(test_arraylist_iterator.getText().get(i));
    }
    System.out.println("\nVerilen String bulundu indexi:"+test_array
    System.out.println();
    */

    System.out.println("\n---LINKLIST ITERATOR IMPLEMENTATION---");
    SimpleTextEditorLinkList test_linklist_iterator=new SimpleTextEc
    test_linklist_iterator.ReadIterator();

    for(int i=0; i<test_linklist_iterator.getText().size(); i++){
        System.out.print(test_linklist_iterator.getText().get(i));
    }
    System.out.println("\n\nAddITerator methodu ile verilen indexe y

    test_linklist_iterator.AddIterator( ekle: "123", index: 6);
    test_linklist_iterator.AddIterator( ekle: "123", index: 1000);
    test_linklist_iterator.AddIterator( ekle: "123", index: 700);
    test_linklist_iterator.AddIterator( ekle: "123", index: 500);

    for(int i=0; i<test_linklist_iterator.getText().size(); i++){
        System.out.print(test_linklist_iterator.getText().get(i));
    }
    System.out.println("\n\nReplaceIterator methodu ile character ar
    test_linklist_iterator.ReplaceIterator( x: 'a', y: 'i');

    for(int i=0; i<test_linklist_iterator.getText().size(); i++){
```

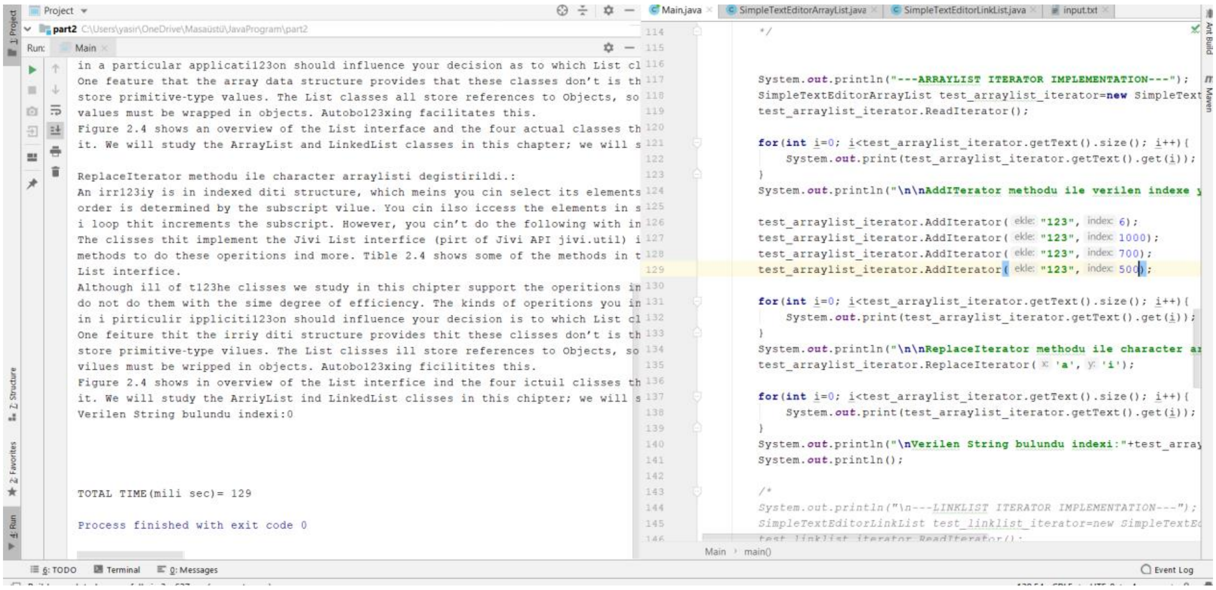
ArrayList Loop


part2 [C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\part2] - ...\src\Main.java [part2] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part2 src Main
Project
part2 C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\part2
Run: Main
Main.java SimpleTextEditorArrayList.java SimpleTextEditorLinkList.java input.txt
Main
1: Project
Z: Structure
2: Favorites
4: Run
Main › main()

```
in a particular applicati123on should influence your decision as to which List cl
One feature that the array data structure provides that these classes don't is th
store primitive-type values. The List classes all store references to Objects, so
values must be wrapped in objects. Autobo123xing facilitates this.
Figure 2.4 shows an overview of the List interface and the four actual classes th
it. We will study the ArrayList and LinkedList classes in this chapter; we will s

ReplaceIterator methodu ile character arraylisti degistirildi.:
An irr123iy is in indexed diti structure, which meins you cin select its elements
order is determined by the subscript vilue. You cin ilso iccess the elements in s
i loop thit increments the subscript. However, you cin't do the following with in
The clisses thit implement the Jivi List interfice (pirt of Jivi API jivi.util) i
methods to do these operitions ind more. Tible 2.4 shows some of the methods in t
List interfice.
Although ill of t123he clisses we study in this chipter support the operitions in
do not do them with the sime degree of efficiency. The kinds of operitions you in
in i pirticulir ippliciti123on should influence your decision is to which List cl
One feiture thit the irriy diti structure provides thit these clisses don't is th
store primitive-type vilues. The List clisses ill store references to Objects, so
vilues must be wripped in objects. Autobo123xing ficilitites this.
Figure 2.4 shows in overview of the List interfice ind the four ictuil clisses th
it. We will study the ArriyList ind LinkedList clisses in this chipter; we will s
Verilen String bulundu indexi:0




TOTAL TIME(mili sec)= 84

Process finished with exit code 0
```

```java
    }
    System.out.println("\n\nReplaceIterator methodu ile character a
    test_linklist_iterator.ReplaceIterator('a','i');

    for(int i=0; i<test_linklist_iterator.getText().size(); i++){
        System.out.print(test_linklist_iterator.getText().get(i));
    }
    System.out.println("\nVerilen String bulundu indexi:"+test_link
    System.out.println();
*/


    System.out.println("\n---ARRAYLIST LOOP IMPLEMENTATION---");
    SimpleTextEditorArrayList test_arraylist_loop=new SimpleTextEdit
    test_arraylist_loop.Read();

    for(int i=0; i<test_arraylist_loop.getText().size(); i++){
        System.out.print(test_arraylist_loop.getText().get(i));
    }
    System.out.println("\n\nAddITerator methodu ile verilen indexe y

    test_arraylist_loop.Add( ekle: "123", index: 6);
    test_arraylist_loop.Add( ekle: "123", index: 1000);
    test_arraylist_loop.Add( ekle: "123", index: 700);
    test_arraylist_loop.Add( ekle: "123", index: 500);

    for(int i=0; i<test_arraylist_loop.getText().size(); i++){
        System.out.print(test_arraylist_loop.getText().get(i));
    }
    System.out.println("\n\nReplaceIterator methodu ile character ar
    test_arraylist_loop.Replace( x: 'a', y: 'i');

    for(int i=0; i<test_arraylist_loop.getText().size(); i++){
        System.out.print(test_arraylist_loop.getText().get(i));
```

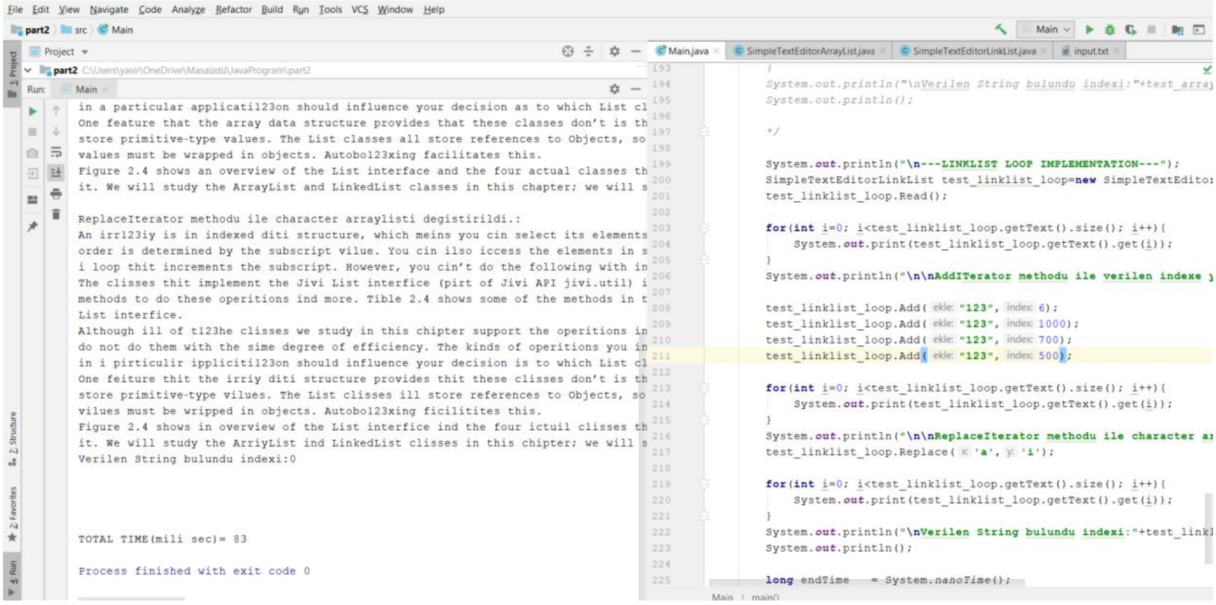
ArrayList Iterator



```
in a particular applicati123on should influence your decision as to which List cl
One feature that the array data structure provides that these classes don’t is th
store primitive-type values. The List classes all store references to Objects, so
values must be wrapped in objects. Autobo123xing facilitates this.
Figure 2.4 shows an overview of the List interface and the four actual classes th
it. We will study the ArrayList and LinkedList classes in this chapter; we will s

ReplaceIterator methodu ile character arraylisti degistirildi.:
An irr123iy is in indexed diti structure, which meins you cin select its elements
order is determined by the subscript vilue. You cin ilso iccess the elements in s
i loop thit increments the subscript. However, you cin’t do the following with in
The clisses thit implement the Jivi List interfice (pirt of Jivi API jivi.util) i
methods to do these operitions ind more. Tible 2.4 shows some of the methods in t
List interfice.
Although ill of t123he clisses we study in this chipter support the operitions in
do not do them with the sime degree of efficiency. The kinds of operitions you in
in i pirticulir ippliciti123on should influence your decision is to which List cl
One feiture thit the irriy diti structure provides thit these clisses don’t is th
store primitive-type vilues. The List clisses ill store references to Objects, so
vilues must be wripped in objects. Autobo123xing ficilitites this.
Figure 2.4 shows in overview of the List interfice ind the four ictuil clisses th
it. We will study the ArriyList ind LinkedList clisses in this chipter; we will s
Verilen String bulundu indexi:0




TOTAL TIME(mili sec)= 129

Process finished with exit code 0
```

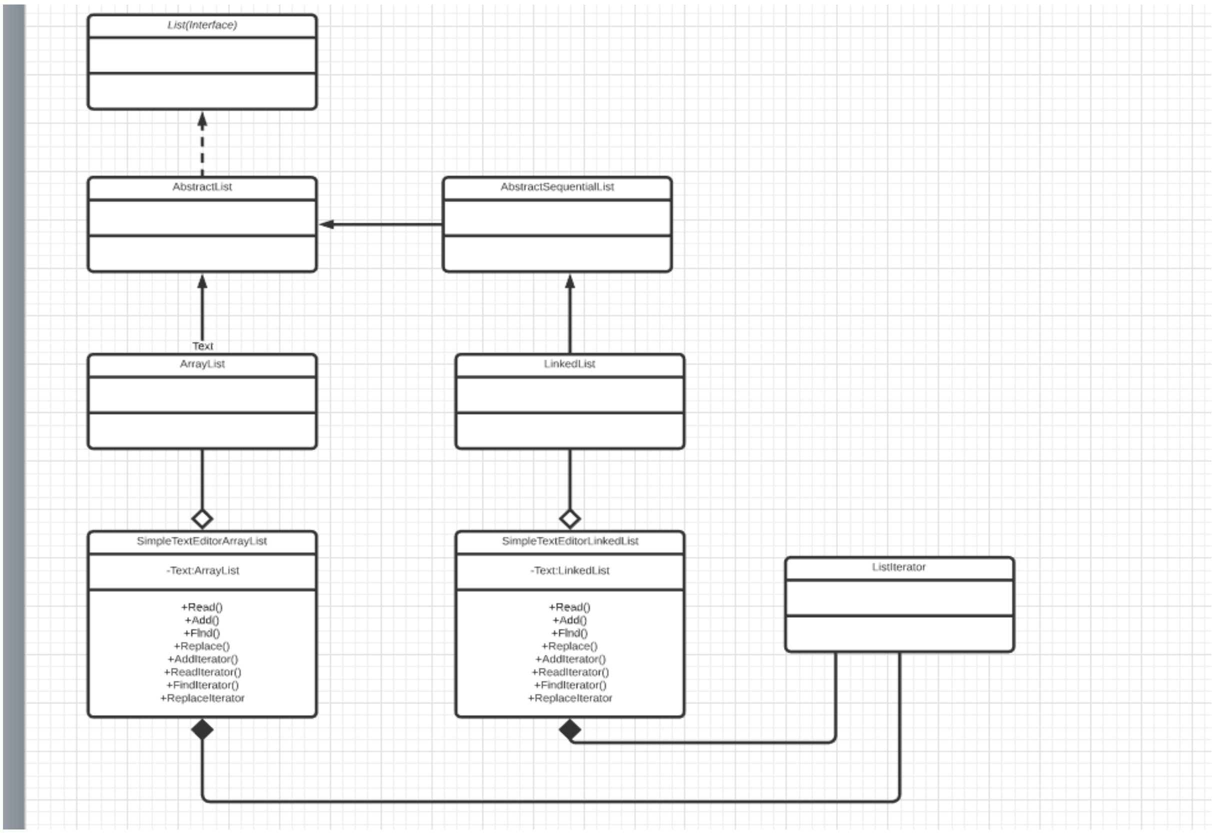
```java
    */


    System.out.println("---ARRAYLIST ITERATOR IMPLEMENTATION---");
    SimpleTextEditorArrayList test_arraylist_iterator=new SimpleText
    test_arraylist_iterator.ReadIterator();

    for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
        System.out.print(test_arraylist_iterator.getText().get(i));
    }
    System.out.println("\n\nAddITerator methodu ile verilen indexe y

    test_arraylist_iterator.AddIterator( ekle: "123", index: 6);
    test_arraylist_iterator.AddIterator( ekle: "123", index: 1000);
    test_arraylist_iterator.AddIterator( ekle: "123", index: 700);
    test_arraylist_iterator.AddIterator( ekle: "123", index: 500);

    for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
        System.out.print(test_arraylist_iterator.getText().get(i));
    }
    System.out.println("\n\nReplaceIterator methodu ile character ar
    test_arraylist_iterator.ReplaceIterator( x: 'a', y: 'i');

    for(int i=0; i<test_arraylist_iterator.getText().size(); i++){
        System.out.print(test_arraylist_iterator.getText().get(i));
    }
    System.out.println("\nVerilen String bulundu indexi:"+test_array
    System.out.println();

    /*
    System.out.println("\n---LINKLIST ITERATOR IMPLEMENTATION---");
    SimpleTextEditorLinkList test_linklist_iterator=new SimpleTextEd
    test_linklist_iterator.ReadIterator();
```

LinkList Loop


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part2 › src › Main
Main
Project
part2 C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\part2
Run: Main
1: Project
Z: Structure
2: Favorites
4: Run
Main.java
SimpleTextEditorArrayList.java
SimpleTextEditorLinkList.java
input.txt
Main › main()

```
in a particular applicati123on should influence your decision as to which List cl
One feature that the array data structure provides that these classes don’t is th
store primitive-type values. The List classes all store references to Objects, so
values must be wrapped in objects. Autobo123xing facilitates this.
Figure 2.4 shows an overview of the List interface and the four actual classes th
it. We will study the ArrayList and LinkedList classes in this chapter; we will s

ReplaceIterator methodu ile character arraylisti degistirildi.:
An irr123iy is in indexed diti structure, which meins you cin select its elements
order is determined by the subscript vilue. You cin ilso iccess the elements in s
i loop thit increments the subscript. However, you cin’t do the following with in
The clisses thit implement the Jivi List interfice (pirt of Jivi API jivi.util) i
methods to do these operitions ind more. Tible 2.4 shows some of the methods in t
List interfice.
Although ill of t123he clisses we study in this chipter support the operitions in
do not do them with the sime degree of efficiency. The kinds of operitions you in
in i pirticulir ippliciti123on should influence your decision is to which List cl
One feiture thit the irriy diti structure provides thit these clisses don’t is th
store primitive-type vilues. The List clisses ill store references to Objects, so
vilues must be wripped in objects. Autobo123xing ficilitites this.
Figure 2.4 shows in overview of the List interfice ind the four ictuil clisses th
it. We will study the ArriyList ind LinkedList clisses in this chipter; we will s
Verilen String bulundu indexi:0




TOTAL TIME(mili sec)= 83

Process finished with exit code 0
```

```java
    }
    System.out.println("\nVerilen String bulundu indexi:"+test_arra
    System.out.println();

    */

    System.out.println("\n---LINKLIST LOOP IMPLEMENTATION---");
    SimpleTextEditorLinkList test_linklist_loop=new SimpleTextEditor
    test_linklist_loop.Read();

    for(int i=0; i<test_linklist_loop.getText().size(); i++){
        System.out.print(test_linklist_loop.getText().get(i));
    }
    System.out.println("\n\nAddITerator methodu ile verilen indexe y

    test_linklist_loop.Add( ekle: "123", index: 6);
    test_linklist_loop.Add( ekle: "123", index: 1000);
    test_linklist_loop.Add( ekle: "123", index: 700);
    test_linklist_loop.Add( ekle: "123", index: 500);

    for(int i=0; i<test_linklist_loop.getText().size(); i++){
        System.out.print(test_linklist_loop.getText().get(i));
    }
    System.out.println("\n\nReplaceIterator methodu ile character ar
    test_linklist_loop.Replace( x: 'a', y: 'i');

    for(int i=0; i<test_linklist_loop.getText().size(); i++){
        System.out.print(test_linklist_loop.getText().get(i));
    }
    System.out.println("\nVerilen String bulundu indexi:"+test_linkl
    System.out.println();

    long endTime   = System.nanoTime();
```



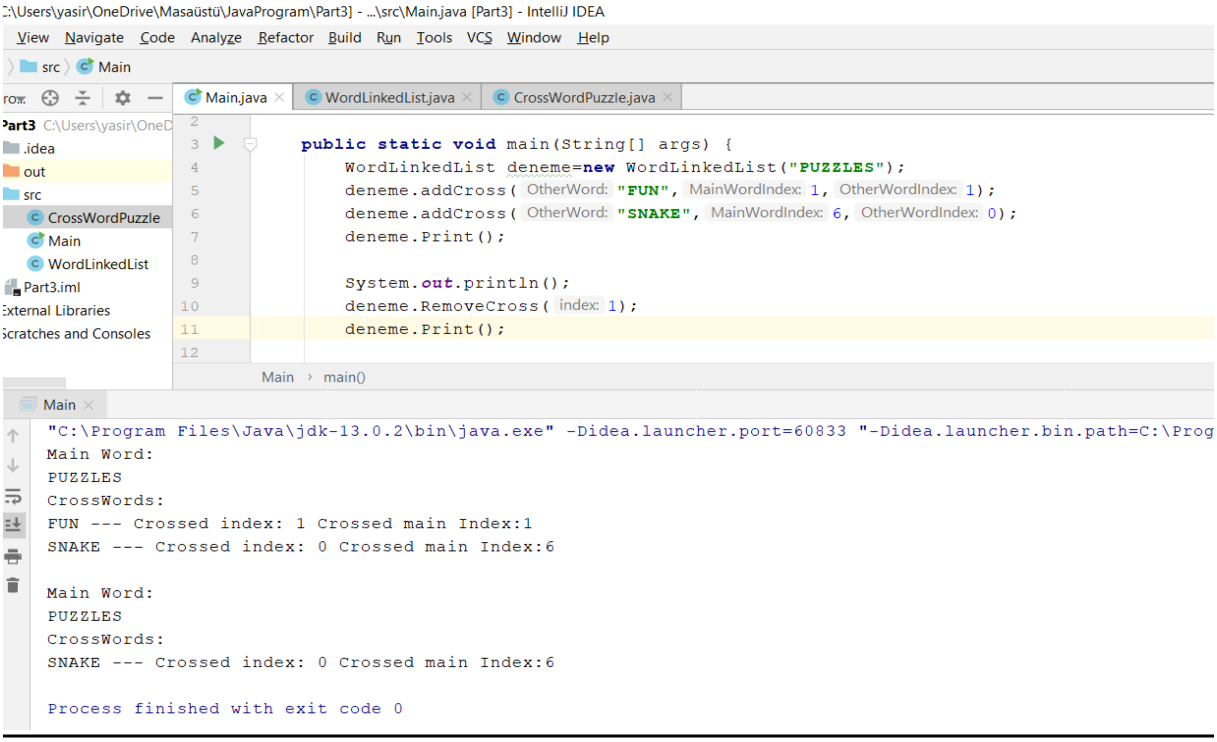
Part2 Class Diagram

PART3

In part3, My first class is WordLinkedList. It is a double linked list but I declare node class 3 referance next,prev and cross. Next points the next node, prev points the previous node and cross referance for referance cross words.With WordLinkedList I hold main word with character Nodes.WordlinkedList can declare this Word with it is one parameter constructer.That constructer takes a string as argument and seperate this string into characters and hold this characters with a list structure.There is 2 add method in this class. First add(addCross) takes a string as argument and adding this string to main word as a cross.

The second add(addMainWord)takes another WordLinkedList object for add another word to the Word like a cross, simply this method like combine to words with their crosses. Like add methods there is 2 remove method. First remove take a index as paramater and remove the cross in this index from word. Second removeAll method, Remove all crosses that Word has.

Print method prints the word, its cross words and cross indexes.


:\Users\yasir\OneDrive\Masaüstü\JavaProgram\Part3] - ...\src\Main.java [Part3] - IntelliJ IDEA
View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src
Main
Main.java
WordLinkedList.java
CrossWordPuzzle.java
Part3 C:\Users\yasir\OneD
.idea
out
src
CrossWordPuzzle
Main
WordLinkedList
Part3.iml
External Libraries
Scratches and Consoles
Main › main()
Main

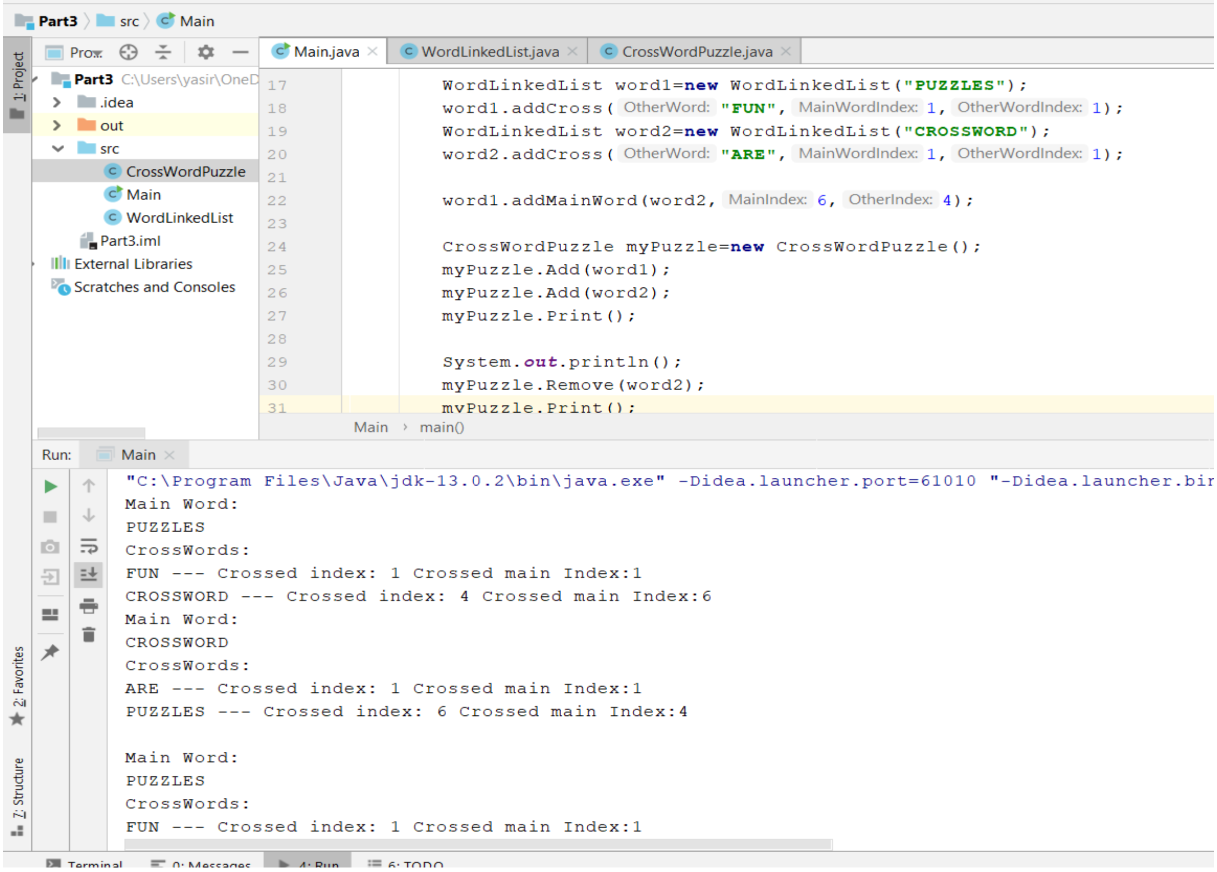
```java
    public static void main(String[] args) {
        WordLinkedList deneme=new WordLinkedList("PUZZLES");
        deneme.addCross( OtherWord: "FUN", MainWordIndex: 1, OtherWordIndex: 1);
        deneme.addCross( OtherWord: "SNAKE", MainWordIndex: 6, OtherWordIndex: 0);
        deneme.Print();

        System.out.println();
        deneme.RemoveCross( index: 1);
        deneme.Print();
```

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" -Didea.launcher.port=60833 "-Didea.launcher.bin.path=C:\Prog
Main Word:
PUZZLES
CrossWords:
FUN --- Crossed index: 1 Crossed main Index:1
SNAKE --- Crossed index: 0 Crossed main Index:6

Main Word:
PUZZLES
CrossWords:
SNAKE --- Crossed index: 0 Crossed main Index:6

Process finished with exit code 0
```

In this test program first I create a WordLinkedList object called deneme. denemes Word is PUZZLES. After that I add 2 cross with using addCross method. First addCross combine PUZZLE index 1(U) and “FUN” index 1(U).Second addCross combine PUZZLE index 6(S) and “SNAKE” index 0(S). Then I call Print method. This Method print Word(PUZZLES) and this words crosses with their connection indexes(FUN and SNAKE).After that print I call remove method with paramater index 1.This method remove the cross related to index 1 in the word(which is “FUN”).Then I call print method again to show “FUN” cross has been deleted.

My Second class is CrossWordPuzzle.This class has a List of WordLinkedList

And 3 method. Add method add Word to list. Remove method remove Word from list and print method Print all word which WordLinkedList has and these words crosses with their indexes.

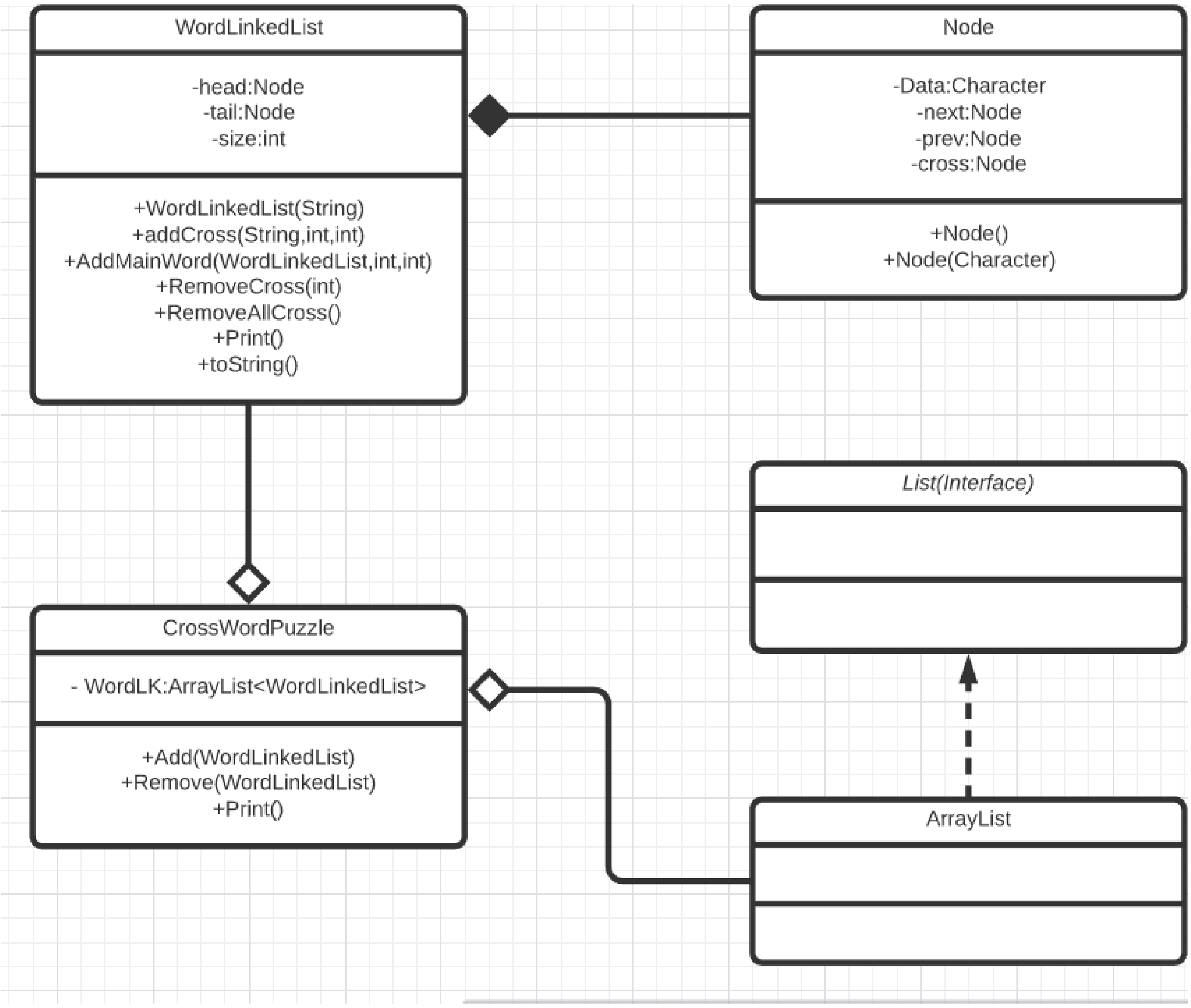


```
        WordLinkedList word1=new WordLinkedList("PUZZLES");
        word1.addCross( OtherWord: "FUN", MainWordIndex: 1, OtherWordIndex: 1);
        WordLinkedList word2=new WordLinkedList("CROSSWORD");
        word2.addCross( OtherWord: "ARE", MainWordIndex: 1, OtherWordIndex: 1);

        word1.addMainWord(word2, MainIndex: 6, OtherIndex: 4);

        CrossWordPuzzle myPuzzle=new CrossWordPuzzle();
        myPuzzle.Add(word1);
        myPuzzle.Add(word2);
        myPuzzle.Print();

        System.out.println();
        myPuzzle.Remove(word2);
        myPuzzle.Print();
```

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" -Didea.launcher.port=61010 "-Didea.launcher.bir
Main Word:
PUZZLES
CrossWords:
FUN --- Crossed index: 1 Crossed main Index:1
CROSSWORD --- Crossed index: 4 Crossed main Index:6
Main Word:
CROSSWORD
CrossWords:
ARE --- Crossed index: 1 Crossed main Index:1
PUZZLES --- Crossed index: 6 Crossed main Index:4

Main Word:
PUZZLES
CrossWords:
FUN --- Crossed index: 1 Crossed main Index:1
```

In this test program, I created 2 WordLinkedList PUZZLES word and CROSSWORD word.I add a cross “FUN” to PUZZLES Word with addCross in index 1. I add a cross “ARE” to CROSSWORD word withaddCross method in index 1. After that with addMainWord method I combine PUZZLES Word and CROSSWORD in PUZZLES index 6(PUZZLE[S]) and CROSSWORD index 4(CROS[S]WORD).

Then I declare a CrossWordPuzzle object and With CrossWordPuzzle class method I add 2 word in list. Then call CrossWordPuzzle print method and it prints the words and their crosses with corresponding indexes.Then I remove word2 with using CrossWordPuzzle remove method.Then again prints CrossWordPuzzle to Show word2 and its crosses has been deleted.



Part 3 Class Diagram