

```
/**  
 * HW03_131044017_Mustafa_Akilli  
 *  
 * File: SpecList  
 *  
 * Description:  
 *  
 * SpecList Class  
 *  
 * @author Mustafa_Akilli  
 * @since Sunday 06 March 2016, 23:32  
 */
```

Projenin Github Linki:

<https://github.com/AKILLIMUSTAFA/CSE22-HW03>

Detailed system requirements

SpecListInt;

* has three extra procedures that perform the following tasks

-Appends all of the elements in the specified collection to the head of the list.

Boolean addAllAtHead(Collection<? extends E> c)

-Finds and returns intersection list (list of unique elements available in both collections)

List<E> getIntersectList (Collection<? extends E> c)

-Sorts and returns list (use cocktail sort algorithm)

List<E> sortList(decreasing or increasing)

SpecList;

* must extend JAVA-LinkedList

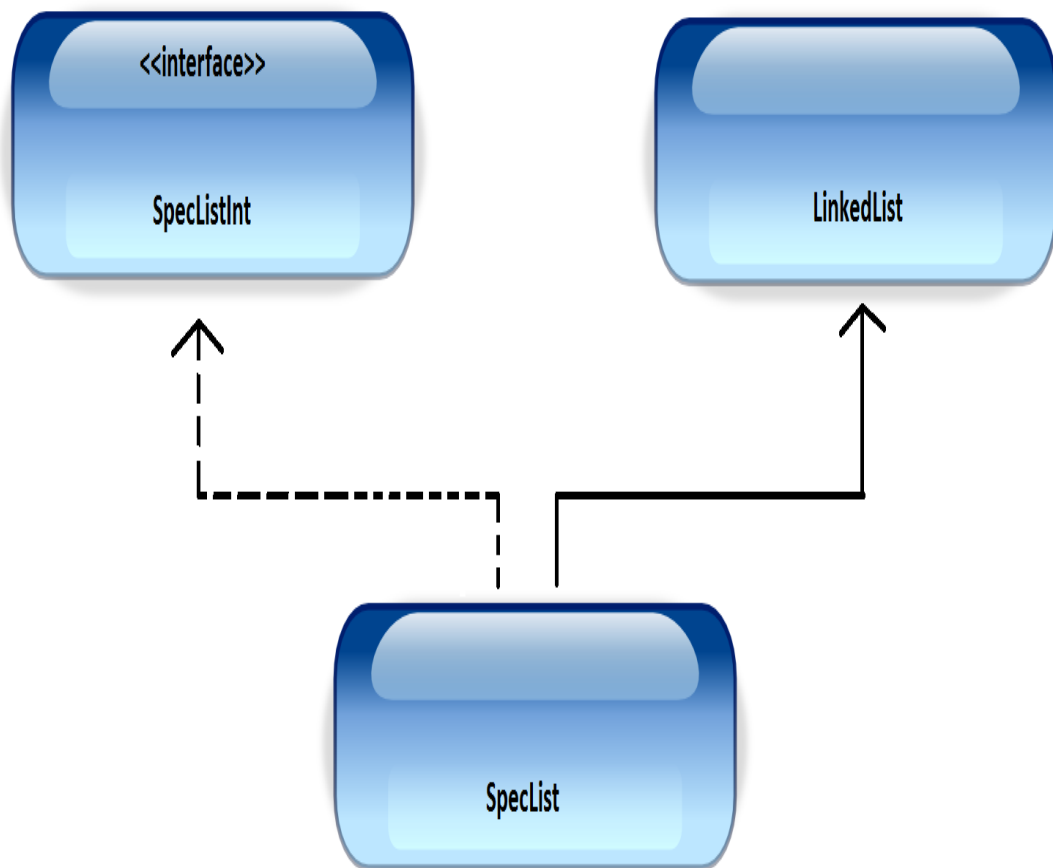
* must implement SpecListInt

The Project Usecase Diagrams

Step	User's Action	System's Response
<u>1</u>	Use addAllAtHead method in SpecList	Appends all of the elements in the specified collection to the head of the list.
<u>2</u>	Use getIntersectList method in SpecList	Finds and returns intersection list (list of unique elements available in both collections)
<u>3</u>	Use sortList method in SpecList	Sorts and returns list (use cocktail sort algorithm)
<u>4</u>	Use sortList method in SpecList	If User a admin, then removing course in system. Otherwise, give a error message.
<u>5</u>	Use add method in SpecList	Appends the specified element to the end of this list.
<u>6</u>	Use addAll method in SpecList	Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator.
<u>7</u>	Use addFirst method in SpecList	Inserts the specified element at the beginning of this list.
<u>8</u>	Use addLast method in SpecList	Appends the specified element to the end of this list.
<u>9</u>	Use clear method in SpecList	Removes all of the elements from this list.
<u>10</u>	Use clone method in SpecList	Returns a shallow copy of this LinkedList.
<u>11</u>	Use contains method in SpecList	Returns true if this list contains the specified element.
<u>12</u>	Use descendingIterator method in SpecList	Returns an iterator over the elements in this deque in reverse sequential order.
<u>13</u>	Use element method in SpecList	Retrieves, but does not remove, the head (first element) of this list.
<u>14</u>	Use get method in SpecList	Returns the element at the specified position in this list.
<u>15</u>	Use getFirst method in SpecList	Returns the first element in this list.
<u>16</u>	Use getLast method in SpecList	Returns the last element in this list.
<u>17</u>	Use indexOf method in SpecList	Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
<u>18</u>	Use lastIndexOf method in SpecList	Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.

<u>19</u>	Use listIterator method in SpecList	Returns a list-iterator of the elements in this list (in proper sequence), starting at the specified position in the list.
<u>20</u>	Use offer method in SpecList	Adds the specified element as the tail (last element) of this list.
<u>21</u>	Use offerFirst method in SpecList	Inserts the specified element at the front of this list.
<u>22</u>	Use offerLast method in SpecList	Inserts the specified element at the end of this list.
<u>23</u>	Use peek method in SpecList	Retrieves, but does not remove, the head (first element) of this list.
<u>24</u>	Use peekFirst method in SpecList	Retrieves, but does not remove, the first element of this list, or returns null if this list is empty.
<u>25</u>	Use peekLast method in SpecList	Retrieves, but does not remove, the last element of this list, or returns null if this list is empty.
<u>26</u>	Use poll method in SpecList	Retrieves and removes the head (first element) of this list.
<u>27</u>	Use pollFirst method in SpecList	Retrieves and removes the first element of this list, or returns null if this list is empty.
<u>28</u>	Use pollLast method in SpecList	Retrieves and removes the last element of this list, or returns null if this list is empty.
<u>29</u>	Use pop method in SpecList	Pops an element from the stack represented by this list.
<u>30</u>	Use push method in SpecList	Pushes an element onto the stack represented by this list.
<u>31</u>	Use remove method in SpecList	Retrieves and removes the head (first element) of this list.
<u>32</u>	Use set method in SpecList	Replaces the element at the specified position in this list with the specified element.
<u>33</u>	Use size method in SpecList	Returns the number of elements in this list.
<u>34</u>	Use toArray method in SpecList	Returns an array containing all of the elements in this list in proper sequence (from first to last element).

Class diagrams



Problem solutions approach

Öncelikle SpecList Class'ını oluşturmak için SpecListInt interface'ini oluşturmaya karar verdim. SpecList interface'i 3 adet method içermektedir.

SpecList Class'ını LinkedList Class'ından extend edip, SpecListInt interface'inide implement ettirdim. Daha Sonra SpecList Classı içindeki 3 methodu implement etmeye başladım.

Bunlar addAllAtHead, getIntersectList ve sortList metodlarıydı. AddAllAtHead metodu parametre olarak aldığı collectionu objenin listesinin başına eklemekte. getIntersectList metodu parametre olarak aldığı collection ile objenin listeninin aynı olan elemanlarının Listini return ediyor. sortList methodu ise parametre olarak aldığı collectionu artan ya da azalan sıra yapıp List tipinde return ediyor.

SpecList class'ı tamamlanınca. Test Senaryosunu ardından jUnit testleri gerçekleştirdim. Daha sonra javadoc dokümantasyonu ve raporu hazırlayarak ödevi sonlandırdım.

Function Notation

addAllAtHead Method:

```
/**
 * Appends all of the elements in the specified collection to the head of the list.
 *
 * @param collection Object
 * @return if the process appends is succesful then return true, otherwise return false
 */
public Boolean addAllAtHead(Collection<? extends E> collection) {
    Iterator<E> iter = (Iterator<E>) collection.iterator(); /*Iterator */
    Boolean checkAnyElementAddToTheHeadOfTheList = false; /*Check Any Element Add To The Head Of The List*/
    Boolean result; /* Result of Appends */

    /*Appends all of the elements in the specified collection to the head of the list.*/
    while(iter.hasNext()){
        addFirst(iter.next());
        checkAnyElementAddToTheHeadOfTheList = true;
    }

    if(checkAnyElementAddToTheHeadOfTheList == true)
        result = true;
    else
        result = false;

    return result;
} //end method addAllAtHead
```



addAllAtHead $\theta(n)$ zamanda çalışır.

hasNext $\theta(1)$ zamanda çalışır.

Next $\theta(1)$ zamanda çalışır.

addFirst $\theta(1)$ zamanda çalışır.

getIntersectList Method:

```
/**
 * Finds and returns intersection list (list of unique elements available in both collections)
 *
 * @param collection Object
 * @return intersection list
 */
public List<E> getIntersectList(Collection<? extends E> collection) {
    List<E> intersectList = new LinkedList<E>(); /*IntersectList*/
    Iterator<E> iter = (Iterator<E>) collection.iterator(); /*Iterator */

    /*Finds and add unique elements into intersection list*/
    while(iter.hasNext()){
        E nextElement = iter.next();
        for(int index=0; index<size(); ++index)
            if(nextElement.equals(get(index))
                if(!intersectList.contains(get(index)))
                    intersectList.add(get(index));
    }

    return intersectList; /* returns intersection list */
} //end method getIntersectList
```

getIntersectList $T_b = O(n^2)$ zamanda çalışır.

getIntersectList $T_w = O(n^3)$ zamanda çalışır.

hasNext $\theta(1)$ zamanda çalışır.

Next $\theta(1)$ zamanda çalışır.

Add $\theta(1)$ zamanda çalışır.

Contains $T_b = O(1)$ zamanda çalışır.

Contains $T_w = O(n)$ zamanda çalışır.

sortList Method:

I used cocktail shaker sort algorithm for sortList.

<u>Worst case performance</u>	$O(n^2)$
<u>Best case performance</u>	$O(n)$
<u>Average case performance</u>	$O(n^2)$
<u>Worst case space complexity</u>	$O(1)$

Running command and results

TESTSCENARIO INTEGER:

Added Elements LinkedList Integer: 5,1,3,2,4

Added Elements SpecList Integer: 22,33,44,55

Appends all of the elements in the LinkedListInteger to the head of the SpecList Integer

Added Elements LinkedList Integer: 66,99,115

Increase Sorted LinkedList Integer for LinkedList Integer

Decrease Sorted LinkedList Integer for LinkedList Integer

Intersection List Integer for specList Integer and linkedList Integer

TESTSCENARIO INTEGER SCREENSHOT:

```
TestScenario (1)
"C:\Program Files\Java\jdk1.8.0_74\bin\java" ...
****TestScenario Integer****

Added Elements LinkedList Integer: 5,1,3,2,4
LinkedList Integer Elements:
5,1,3,2,4,
*****

Added Elements SpecList Integer: 22,33,44,55
SpecList Integer Elements:
5,4,44,55,
*****

Appends all of the elements in the LinkedListInteger to the head of the SpecList Integer
SpecList Integer Elements:
4,2,3,1,5,5,4,44,55,
*****

Added Elements LinkedList Integer: 66,99,115
LinkedList Integer Elements:
5,1,3,2,4,66,99,115,
*****

Increase Sorted LinkedList Integer for LinkedList Integer
Increase Sorted LinkedList Integer Elements:
1,2,3,4,5,66,99,115,
*****

Decrease Sorted LinkedList Integer for LinkedList Integer
Decrease Sorted LinkedList Integer Elements:
115,99,66,5,4,3,2,1,
*****

Intersection List Integer for specList Integer and linkedList Integer
Intersection List Integer Elements:
5,1,3,2,4,|
*****
```

TESTSCENARIO DOUBLE:

Added Elements LinkedList Double: 4.8, 3.2, 5.8, 7.8, 3.7

Added Elements SpecList Double: 22.8,33.6,44.4,55.7

Appends all of the elements in the LinkedListDouble to the head of the SpecList Double

Added Elements LinkedList Double: 115.7,99.9,58.0

Increase Sorted LinkedList Double for LinkedList Double

Decrease Sorted LinkedList Double for LinkedList Double

Intersection List Double for specList Double and linkedList Double

TESTSCENARIO DOUBLE SCREENSHOT:

```
TestScenario (1)

****TestScenario Double****

Added Elements LinkedList Double: 4.8, 3.2, 5.8, 7.8, 3.7
LinkedList Double Elements:
4.8,3.2,5.8,7.8,3.7,
*****

Added Elements SpecList Double: 22.8,33.6,44.4,55.7
SpecList Double Elements:
22.8,33.6,44.4,55.7,
*****

Appends all of the elements in the LinkedListDouble to the head of the SpecList Double
SpecList Double Elements:
3.7,7.8,5.8,3.2,4.8,22.8,33.6,44.4,55.7,
*****

Added Elements LinkedList Double: 115.7,99.9,58.0
LinkedList Double Elements:
4.8,3.2,5.8,7.8,3.7,115.7,99.9,58.0,
*****

Increase Sorted LinkedList Double for LinkedList Double
Increase Sorted LinkedList Double Elements:
3.2,3.7,4.8,5.8,7.8,58.0,99.9,115.7,
*****

Decrease Sorted LinkedList Double for LinkedList Double
Decrease Sorted LinkedList Double Elements:
115.7,99.9,58.0,7.8,5.8,4.8,3.7,3.2,
*****

Intersection List Double for specList Double and linkedList Double
Intersection List Double Elements:
4.8,3.2,5.8,7.8,3.7,
*****
```

TESTSCENARIO CHARACTER:

Added Elements LinkedList Character: e,a,c,d,b

Added Elements SpecList Character: t,u,v,y

Appends all of the elements in the LinkedListCharacter to the head of the SpecList Character

Added Elements LinkedList Character: q,x,w

Increase Sorted LinkedList Character for LinkedList Character

Decrease Sorted LinkedList Character for LinkedList Character

Intersection List Character for specList Character and linkedList Character

TESTSCENARIO CHARACTER SCREENSHOT:

```
TestScenario (1)

*****
****TestScenario Character****

Added Elements LinkedList Character: e,a,c,d,b
LinkedList Character Elements:
e,a,c,d,b,
*****

Added Elements SpecList Character: t,u,v,y
SpecList Character Elements:
t,u,v,y,
*****

Appends all of the elements in the LinkedListCharacter to the head of the SpecList Character
SpecList Character Elements:
b,d,c,a,e,t,u,v,y,
*****

Added Elements LinkedList Character: q,x,w
LinkedList Character Elements:
e,a,c,d,b,q,x,w,
*****

Increase Sorted LinkedList Character for LinkedList Character
Increase Sorted LinkedList Character Elements:
a,b,c,d,e,q,w,x,
*****

Decrease Sorted LinkedList Character for LinkedList Character
Decrease Sorted LinkedList Character Elements:
x,w,q,e,d,c,b,a,
*****

Intersection List Character for specList Character and linkedList Character
Intersection List Character Elements:
e,a,c,d,b,
*****
```

Test Case

JUNIT TESTS:

testAddAllAtHead Method:

I created a firstList (SpecList<Integer>), secondList (SpecList<Integer>) and secondListPlusFirstList (SpecList<Integer>). Added elements (1,2,3) into firstList. Added elements (5,6,7) into secondList. Added elements (7,6,5,1,2,3) into secondListPlusFirstList. And I expect a true result for: firstList.addAllAtHead(secondList) have same elements like secondListPlusFirstList.

testAddAllAtHeadSecond Method:

I created a firstList (SpecList<Integer>), secondList (SpecList<Integer>) and secondListPlusFirstList (SpecList<Integer>). Added elements (1,2,3) into firstList. Added elements (5,6,7) into secondList. Added elements (8,9,10,1,2,3) into secondListPlusFirstList. And I expect a false result for: firstList.addAllAtHead(secondList) have not same elements like secondListPlusFirstList.

testGetIntersectList Method:

I created a firstList (SpecList<Integer>), secondList (SpecList<Integer>) and secondListPlusFirstList (SpecList<Integer>). Added elements (1,2,3,4,5) into firstList. Added elements (4,5,6,7) into secondList. Added elements (4,5) into secondListPlusFirstList. And I expect a true result for: firstList.getIntersectList(secondList) have same elements like secondListPlusFirstList.

testGetIntersectListSecond Method:

I created a firstList (SpecList<Integer>), secondList (SpecList<Integer>) and secondListPlusFirstList (SpecList<Integer>). Added elements (1,2,3,4,5) into firstList. Added elements (4,5,6,7) into secondList. Added elements (8,5) into secondListPlusFirstList. And I expect a false result for: firstList.getIntersectList(secondList) have not same elements like secondListPlusFirstList.

testSortList Method:

I created a testList (SpecList<Integer>), firstList (SpecList<Integer>), sortedFirstList (List<Integer>) and sortedList (SpecList<Integer>). Added elements (10,8,15,7,9) into firstList. Added elements (7,8,9,10,15) into sortedList. And I expect a true result for: sortedFirstList = testList.sortList(firstList,SpecList.INCREASING) have same elements like sortedList.

testSortListSecond Method:

I created a testList (SpecList<Integer>), firstList (SpecList<Integer>), sortedFirstList (List<Integer>) and sortedList (SpecList<Integer>). Added elements (10,8,15,7,9) into firstList. Added elements (25,65,9,11,15) into sortedList. And I expect a false result for: sortedFirstList = testList.sortList(firstList,SpecList.INCREASING) have not same elements like sortedList.

JUNIT SCREENSHOT:

```
hw03 [-Dtest=SpecListTest, test]
[INFO] Building hw03 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ hw03 ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\MustafaMonster\Desktop\CSE222 ODEVLER\CSE222-HW03\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ hw03 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @ hw03 ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\MustafaMonster\Desktop\CSE222 ODEVLER\CSE222-HW03\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ hw03 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ hw03 ---
[INFO] Surefire report directory: C:\Users\MustafaMonster\Desktop\CSE222 ODEVLER\CSE222-HW03\target\surefire-reports

-----
T E S T S
-----

Running tr.edu.gtu.mustafa.akilli.cse222.SpecList.SpecListTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.012 sec

Results :

Tests run: 6, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.679s
[INFO] Finished at: Tue Mar 08 19:37:15 EET 2016
[INFO] Final Memory: 8M/245M
```