Monday April 1

Lecture 23

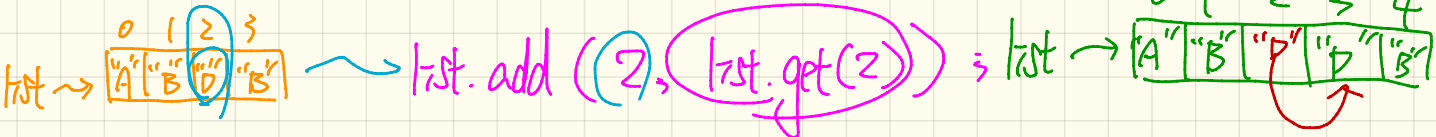
Lab 7 (not to be graded no submission)

§

API

M, T, W no attendance of lab sessions

(W)



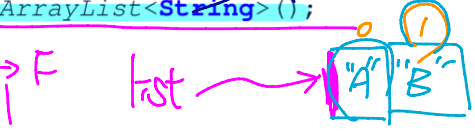

API: ArrayList

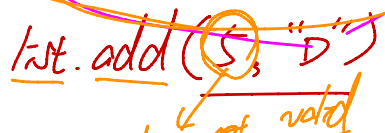
| | |
|---|---|
| int | **size()**<br>Returns the number of elements in this list. |
| boolean | **add(E e)**<br>Appends the specified element to the end of this list. |
| void | **add(int index, E element)**<br>Inserts the specified element at the specified position in this list. |
| boolean | **contains(Object o)**<br>Returns true if this list contains the specified element. |
| E | **remove(int index)**<br>Removes the element at the specified position in this list. |
| boolean | **remove(Object o)**<br>Removes the first occurrence of the specified element from this list, if it is present. |
| int | **indexOf(Object o)**<br>Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element. |
| E | **get(int index)**<br>Returns the element at the specified position in this list. |

when duplicates exist

# Use of ArrayList

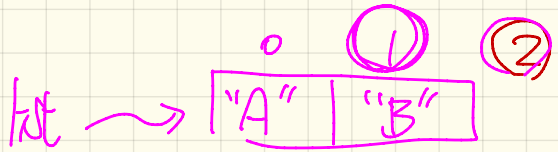

```
ArrayList<String> list = new ArrayList<String>();
println(list.size());
println(list.contains("A"));
println(list.indexOf("A"));
list.add("A");
list.add("B");
println(list.contains("A")); println(list.contains("B")); println(list.contains("C"));
println(list.indexOf("A")); println(list.indexOf("B")); println(list.indexOf("C"));
list.add(1, "C");
println(list.contains("A")); println(list.contains("B")); println(list.contains("C"));
println(list.indexOf("A")); println(list.indexOf("B")); println(list.indexOf("C"));
list.remove("C");
println(list.contains("A")); println(list.contains("B")); println(list.contains("C"));
println(list.indexOf("A")); println(list.indexOf("B")); println(list.indexOf("C"));

for(int i = 0; i < list.size(); i ++) {
  println(list.get(i));
}
```





ArrayList<String> list = new ArrayList<>();




add to end "D"

(Approach 1)

list.add("D") void add(int index, element)

insert to the beginning: list.add(0, __)

insert to the middle: list.add(i, __)

i = 1, 2, ..., list.size()-1

list.add(i, __)

i → list.size() exception.

insert to the end: list.add(list.size(), __)

| | |
|---|---|
| int | **size**()<br>Returns the number of keys in this hashtable. |
| boolean | **containsKey**(**Object** key)<br>Tests if the specified object is a key in this hashtable. |
| boolean | **containsValue**(**Object** value)<br>Returns true if this hashtable maps one or more keys to this value. |
| V | **get**(**Object** key)<br>Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. |
| V | **put**(**K** key, **V** value)<br>Maps the specified key to the specified value in this hashtable. |
| V | **remove**(**Object** key)<br>Removes the key (and its corresponding value) from this hashtable. |

API: HashTable

# Hash table

a collection of entries
- key
- value