Khubayb Kapila
AEE640
JDK 14
@00553883

Data structures and algorithms

Khubayb Kapila

Using JDK 14

Assignment 1

11/30/20

Khubayb Kapila
AEE640
JDK 14
@00553883

Justification of ADT used

One of the main reasons as to why I used an array List is that it allows us to search for Items faster. This is a positive as in a to do list, we require to search through the index of the array to remove the task they are searching for. Time complexity in regards to calling get index method, the complexity time is O(1). If I was to use linked list, it would require me to traverse the full array in order to find the correct index and the complexity for this is O(n).

Additionally, when we add a task that we have to complete into the ArrayList, we initially check the size of the current array and if it is full, we create a new spot for it by copying over the array. The complexity when we don't need to add an item is O(1) and when we do copy and item over, it is O(n). When adding the linked list, the complexity would be O(n) as we have to traverse the array and look for the null position and add it to it until the index has been met.

Moreover, arrays are dynamic. This means they do not hold a definite size. This is a positive as this application can gave the ability to not take up too much memory. The more memory that is being utilized would put pressure on the system being used and this could affect the overall speed of the way the application runs.

When adding items to a list, they can be inserted and deleted from particular positions and also manipulate the data that is stored in there.

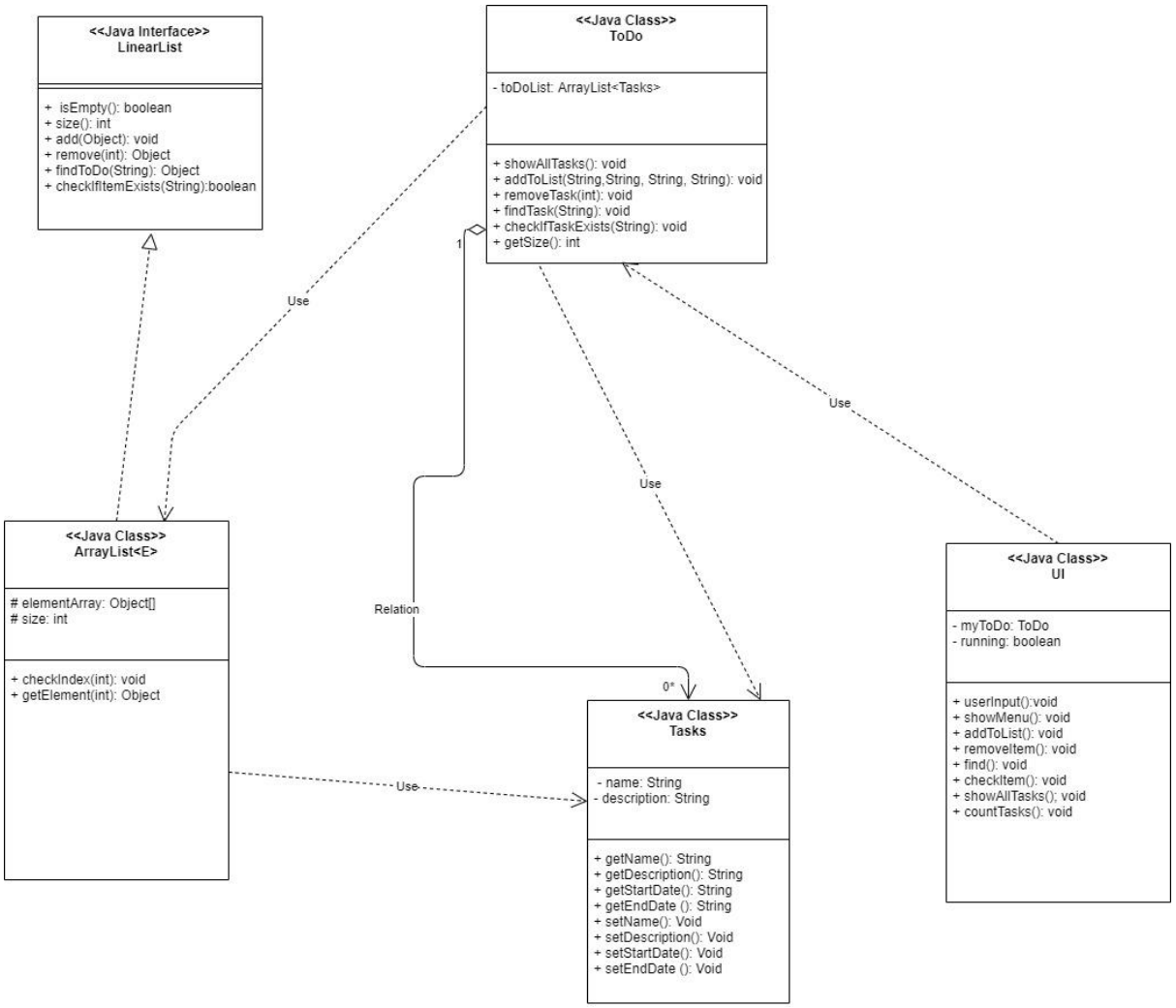
Overall compared to LinkedList, there are positives to both sides however in my opinion if we take into consideration the time complexities, on an overall scale array List is O(n) and as I search, add and remove from my array List, this seemed like the best data structure to develop and use compared to linked list as they are best for inserting items at the front and end of the list.

Java libraires used

- `java.util.Date;`
- `java.lang.IllegalArgumentException;`
- `java.util.Scanner;`

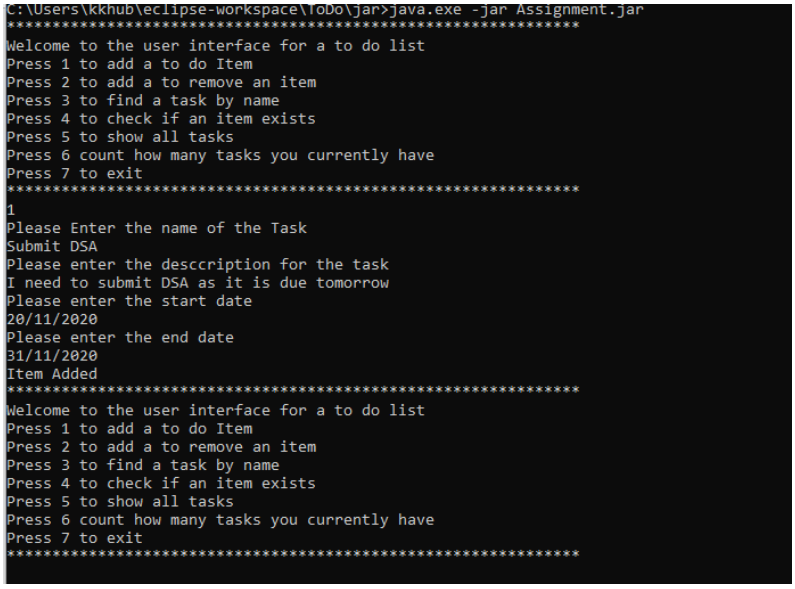
Khubayb Kapila
AEE640
JDK 14
@00553883

Class diagram:

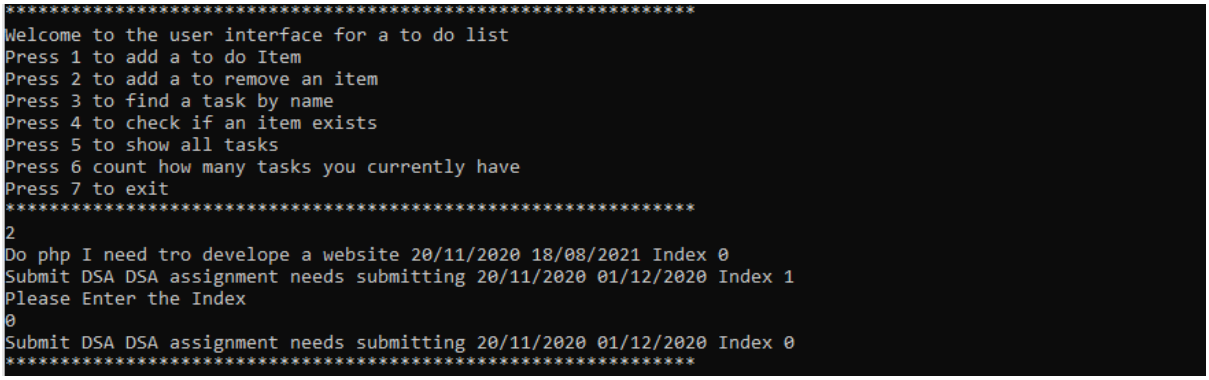


Khubayb Kapila
AEE640
JDK 14
@00553883

Screenshots:



This screen shot shows us that I can successfully add an item to the array List. Initially they are given the menu option to show them the different inputs.



This next screenshot shows us that the user is able to remove an item. Initially, there has to be items in the list for it to work as it will output no items in the list however as you can see it first shows the user all the items in the list then prompts us for the index which is shown on the right hand side of the task. The user then enters an index and when they remove an item, it outputs the current values in the list with their position.

Khubayb Kapila
AEE640
JDK 14
@00553883

```
Submit DSA DSA assignment needs submitting 20/11/2020 01/12/2020 Index 0
**********
```

The image above is simply just showing the position of the item for correctness.

```
**********
Welcome to the user interface for a to do list
Press 1 to add a to do Item
Press 2 to add a to remove an item
Press 3 to find a task by name
Press 4 to check if an item exists
Press 5 to show all tasks
Press 6 count how many tasks you currently have
Press 7 to exit
**********
3
Please enter the name of the task you are looking for
submit dsa
Position: 0
**********
```

This image is option 3. Option 3 is the ability to find the task by name. As we can see in the image above the index for “Submit DSA” is 0. Also note that it has uppercase letters in it. I then enter option 3 and type in “submit dsa” (With no capitals) and it still finds it as I have an .toLowerCase involved in my code. This then outputs the correct position of the value.

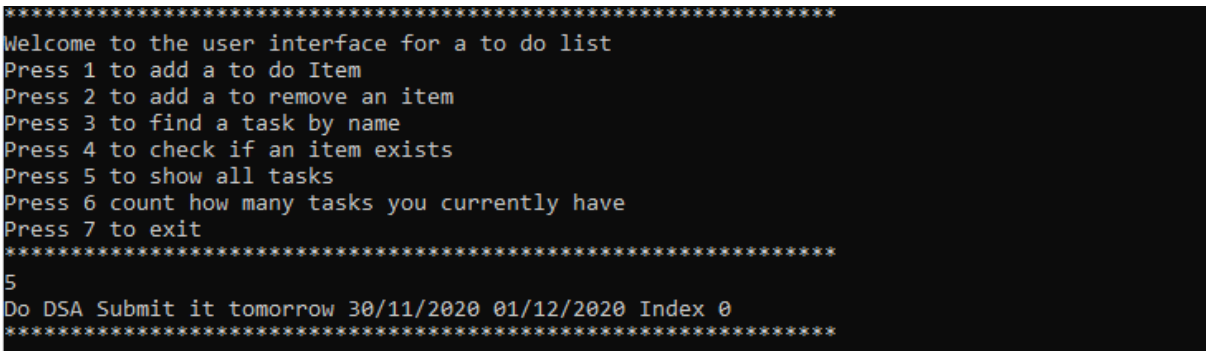
```
**********
1
Please Enter the name of the Task
Do DSA
```

I have now added a task called “Do DSA”.

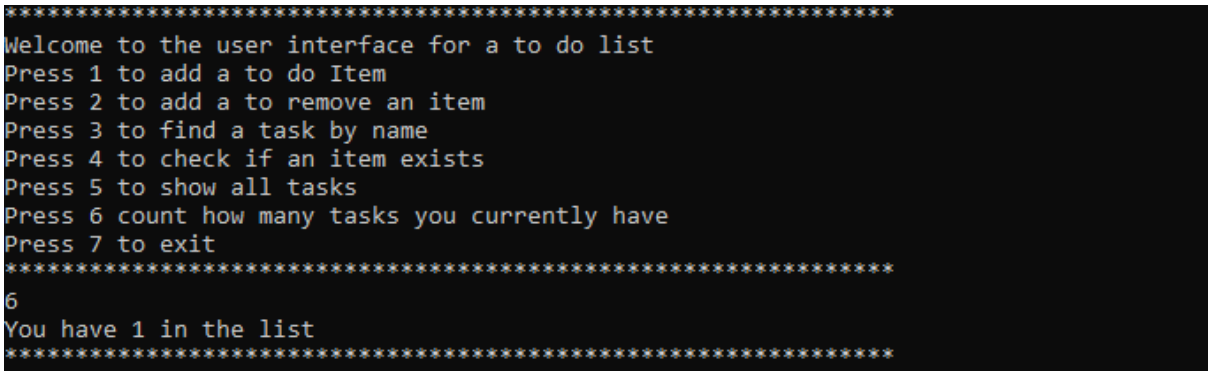
```
**********
Welcome to the user interface for a to do list
Press 1 to add a to do Item
Press 2 to add a to remove an item
Press 3 to find a task by name
Press 4 to check if an item exists
Press 5 to show all tasks
Press 6 count how many tasks you currently have
Press 7 to exit
**********
4
Please enter the name of the task you are looking for
do dsa
Item exists
**********
```

I have searched to see if an item exists, using again lowercase to see if it works and it does. This again utilises the .toLowerCase method.

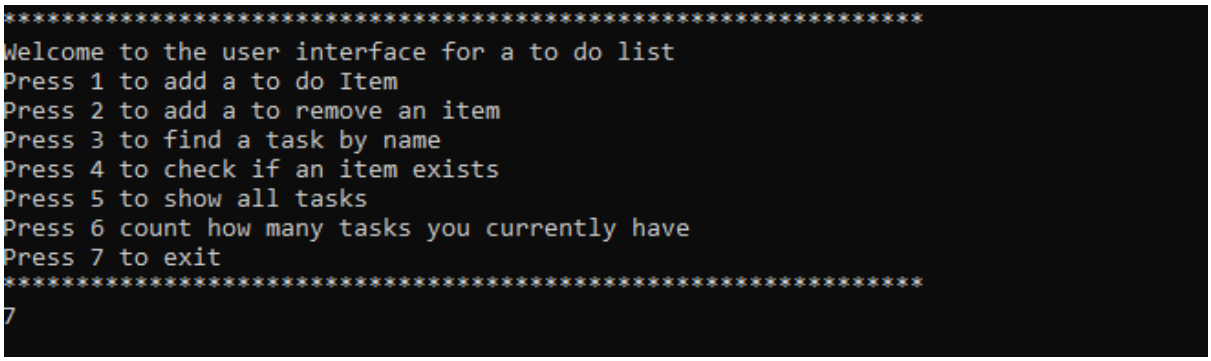
Khubayb Kapila
AEE640
JDK 14
@00553883



Option 5 gives the user the ability to show all tasks that are stored



Option 6: This counts how many tasks are in the array



Option 7: This enables me to exit the loop

Sources : Data Structures and Algorithms in Java
ISBM : 978-1-118-80857-3
Author – Micheal T. Goodrich, Roberto Tamassia, Michael H. Goldwasser