INFO1113/COMP9003   Week 4 Tutorial

Collections

Java has a comprehensive collection library, being able to fit many different requirements for your application. These collections allow us to organise data, store data efficiently and improve readability. You may have encountered the data structure type `ArrayList` already. `List` types store data sequentially and each element is ordered from 0 to $N - 1$. To access elements in a list, Java allows access of elements through indexing, similarly to Arrays.

`ArrayList` is similar to the `list` data structure in python. We can append, insert, set and retrieve elements from the list. One of the main advantages over Array is that it will automatically resize and proves an easier method for handling an unknown number of inputs.

ArrayList

`ArrayList`, as the name infers, contains an array that is resized once the capacity has been reached. `ArrayList`s allows storage of `String`, `Integer` and even `ArrayList`. However you cannot store primitive type objects.

Example:

```java
ArrayList<String> list = new ArrayList<String>();
list.add("String 1");
list.add("Second String");

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

Refer to the ArrayList documentation.

Discussion

- What is a primitive type object? Can you list some examples of primitive types?

- Try creating an ArrayList with a primitive type. What happens?

Question 1: Logger

You are to write a program that will log all the lines that a user has written and store them in a `List`. Once no input has been provided, the program should output each line to the user.

```
Scanner scan = new Scanner(System.in);
//Define your List variable here

while(/* use hasNextLine() */ ) {
        //Your code
}
```

```
1. It was a bright cold day in August.
2. Life is beautiful
                - rest of the output snipped-
```

Extension: Expand your program to create a new `Log` when the user inputs `END LOG` in their program. If no input is given, the final output should specify the different log entries.

Discussion: My Types!

The java compiler will notify you if you are performing an `Unchecked or unsafe operation`. Discuss with your peers the implications of an unchecked or unsafe operation.

After the discussion, **test** yourself by inspecting the code below and specify which a code segments may perform an unchecked operation.

Code 1:

```
ArrayList<String> list = new ArrayList<String>();
list.add("Hello There!");
```

Code 2:

```
ArrayList list = new ArrayList();
list.add("Hello There!");
```

Code 3:

```
ArrayList list = new ArrayList<String>();
list.add("Hello There!");
```

Maps

Map data structures, as inferred by their name, allow data to *map* another set of data. Each entry within the data structure has a *key* that maps to a *value*. These are commonly referred to as Key-Value pairs.

In the following example we have a `HashMap` that maps `String` objects to `Integer` objects.

```
Map<String, Integer> map = new HashMap<String, Integer>();
```

A critical difference between list types and map types are that the keys in a map are unique, as in you cannot have two keys that evaluate to be equivalent.

For example, we are able to create an entry with the key `"Five"` and associated it with the value `5`.

```
map.put("Five", new Integer(5));
```

Question 2: Character Occurrence

Write a program which allow user to keep entering strings. The program will update and display the cumulative occurrence of each character in all previous strings user input. User can press ctrl + d in the terminal to terminate the program.

You should use a `HashMap` to store relevant data.

```
Please input a string: Hello
{e=1, o=1, l=2, H=1}
Please input a string: Haha
{e=1, a=2, o=1, l=2, H=2, h=1}
```

Question 3: Roll Call

You are the owner of a pet day care and have to keep track of the different animals coming in and out. Users enter whether the pet is entering or leaving the day care, the type of pet and the pet's name. After each entry print out the number of animals in the day care. Users enter their pet as `<In/Out>` `<Pet Type>` `<Pet Name>`

Example:

```
Please sign in/out your pet: In Dog Spot
{Dog: 1}
Please sign in/out your pet: In Cat Tiggs
{Dog: 1, Cat: 1}
Please sign in/out your pet: Out Dog Spot
{Cat: 1}
```

Question 4: Colour Word to Colour Object

Create a program which will associate a `String` to a `Colour` object. The keys should represent the colour as an english word while the value must contain Red-Green-Blue integer values.

`Blue -> 0, 0, 255`

Use a `HashMap` to store each entry. Some colours to work with:

- Red: 255, 0, 0

- Purple: 120, 0, 200

- Grass Green: 0, 150, 50

- Brown: 150, 100, 50

Extension: Implement an `add()` method to colour which will produce a new colour from the addition two colour objects. (Each colour channel value must not exceed 255).

Question 5: Building your own arraylist

We can build our own data structures and are not limited to what the java standard library provides. Sometimes we may want a data structure which is tuned to our scenario. In this question we will be building our own `ArrayList` data structure called `DynamicArray`.

```java
public class DynamicArray {

        private int[] array;
        private int size;


        public DynamicArray()

        /**
         * @param index gets the value from a specific index
         * @return element stored at index
         */
        public int get(int index)

        /**
         * Adds an element at the end of the list
         * @param element to be stored at size
         */
        public void add(int element)

        /**
         * @param returns the size of the array
         */
        public int size()

}
```

Question 6: Assessed Task: Online Task 3

Remember you are required to complete a Online Task within the due date. Go to EdStem for this unit and click on Lessons to find out the task and the due date. This is a marked task. Note that you are allowed to submit multiple times but only the last one will be marked.