Loops

BNTA

Summary

- Java has a something called a `Collections` framework
- Everything in this framework inherits* from the Collections interface*
  - *more on this later!
- What this means for us is that we have a load of ‘in-built’ methods we can use for every collection type
- Focussing on 2 main data structures (in terms of ‘collection of objects’): Arrays and ArrayLists
- ArrayLists are probably of most interest to us
- Key feature of ArrayLists is that they can grow in size
  - (Mary Poppins bag)

Disclaimer

If I say ‘array’ I mean ArrayList.

What are loops?

One of the points of having something in a collection is so we can loop over it

Loops execute a block of code a specified number of times.

We want to write ‘Clean Code’.

Code that doesn’t repeat itself (or DRY).

Don’t **R**epeat **Y**ourself

Instead of repeating ourselves over and over again, we can tell Java to run it multiple times.

1. **Enhanced for loop**
2. **‘Classic’ for loop**

Need to know: Increment and Decrement Operators

- `++` is used to increment (increase) the value of a variable
- `--` is used to decrement (decrease) the value of a variable
- Where they are positioned in (before or after the variable) slightly changes the behaviour

(AFTER the variable returns value then increments)

```
int i = 0;

System.out.println(i++)
// 0

System.out.println(i)
// 1
```

(BEFORE the variable Increments first then returns value)

```
int i = 0;

System.out.println(++i)
// 1

System.out.println(i)
// 1
```

Enhanced for loop

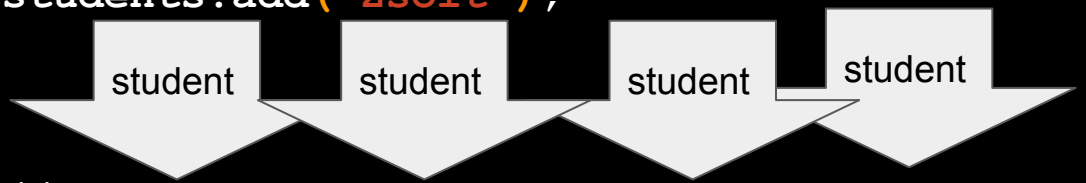
- Most common scenario: we want to perform an action/computation/function on every item in the ArrayList.
- The enhanced for loop has the advantage of being more readable

```
for (String name : names){
    System.out.println(name)
}
```

For every element in the names array, print out that name

String student : students

```
ArrayList<String> students = new ArrayList<>();
students.add("Anna");
students.add("Richard");
students.add("Colin");
students.add("Zsolt");
```




```
// ["Anna", "Richard", "Colin", "Zsolt"]

for (String student : students){
    System.out.println(student);
}
```

```
// output

Anna
Richard
Colin
Zsolt
```

The Classic For Loop

- Not as readable
- Can be intimidating
- Advantage: can be more flexible than the enhanced for loop

```
for (int i = 1; i <= 5; i++){
        System.out.println(i)
    };
```

```
//
1 // i == 1; true; i = 2
2 // i == 2; true; i = 3
3 // i == 3; true; i = 4
4 // i == 4; true; i = 5
5 // i == 5; true; i = 6
// i == 6; false
```

The initial expression. Initialises the starting value of `i`.

The stop condition. In what conditions do you want the loop to carry on for? When this expression evaluates to `false`, the loop will stop.

Updates the value of `i` for each iteration.