

Essential Collections

Java Developer

StayAhead; April, 2022

Java Collections

Essential Collections

- An array is a fixed-size, ordered collection which can store values and refs
- A List is a variable-size, ordered collection which can store refs only
- A Map is a variable-size, unordered collection of key value pairs which can store refs only
- Use an array when the collection need not grow and/or shrink and you need not access individual elements (all elements will be processed)
- Use a List when the collection must grow and/or shrink
- Use a Map when you need to access individual elements

Generic Classes

Essential Collections

- A generic class is one whose instances can deal with a variety of data types whilst maintaining type safety
- All Java's Collection classes are generic - this means you must specify what type of thing you intend to store in the Collection when you create it
- That is, the class is generic but the instance is specific

```
// this ArrayList instance can store Strings only  
var myListOfStrings = new ArrayList<String>();
```

- Restricting, to one, the type of thing that can be stored makes it easier to process the elements (no type-checking is necessary)

Generic Classes

Essential Collections

- In the docs ArrayList is defined like this: `ArrayList<T>`
- The T is a type parameter (T for type)
- T will be erased and replaced with a real type at instantiation time, e.g.

```
var names = new ArrayList<String>();
```

- T is not the only letter used for type parameters
- In the docs HashMap is defined like this: `Map<K, V>`

Arrays Review

Essential Collections

- An array is an object/a collection of values/references of the same type
- Arrays are indexed (ordered) with the first element at index 0
- Arrays are fixed-size; they cannot grow nor shrink
- Array elements may be overwritten

Arrays

Essential Collections

- Array creation may take one of three forms, e.g.

```
// an array of 5 Strings; elements are null by default  
var names = new String[5];
```

```
// an array of 5 ints  
var oddNums = new int[] {1, 3, 5, 7, 9};
```

```
// ditto  
int[] evenNums = {2, 4, 6, 8, 10};
```

Arrays

Essential Collections

- Getting and setting array elements requires the specification of the index inside square brackets, e.g.

```
// getting  
var secondName = names[1];
```

```
// setting  
names[1] = "Thomas";
```

Arrays

Essential Collections

- Every array object has a **length** field
- Array objects have no methods other than those inherited from Object
- An attempt to read from/write to a non-existent array index will result in an exception at runtime
- An enhanced for loop is/was the common method of traversing an array
- Since Java 8, Streams are commonly used to iterate over collections, particularly when performing transformations

Arrays

Essential Collections

- Multi-dimensional arrays are supported, e.g.

// each array element references another array

```
var gameBoard = new char[3][3];
```

```
gameBoard[1][1] = 'X';
```

```
gameBoard[0][2] = '0';
```

```
gameBoard[0][1] = 'X';
```

```
...
```

- There is no limit to the number of dimensions

Arrays

Essential Collections

- The main method has one parameter - an array of Strings

```
public static void main(String[] args) { ... }
```

- The JVM populates this array with the arguments input by the user when starting the app, if any, e.g.

```
$> java App arg1 arg2
```

ArrayList

Essential Collections

- ArrayList is probably the most commonly used of the Collection classes
- It, like all Collection classes, exists in the java.util package
- An ArrayList instance is backed by an array but can grow and shrink
- ArrayList elements may be duplicated
- Elements may be appended, inserted, and removed
- Getting and setting elements requires the specification of an index
- Use an ArrayList when you need not access individual elements but rather when you intend to process all elements

ArrayList

Essential Collections

// instantiation

```
var names = new ArrayList<String>();
```

// appending

```
names.add("Tom");
```

// membership testing

```
var tomIsPresent = names.contains("Tom");
```

// sizing

```
var numElements = names.size();
```

ArrayList

Essential Collections

// getting

```
var firstName = names.get(0);
```

// setting (1st param = index, 2nd param = new value)
names.set(0, "Thomas");

// removing

```
var thomasWasPresent = names.remove("Thomas");
```

// iterating

```
for (var name : names) { ... }
```

HashMap

Essential Collections

- HashMap is probably the most commonly used of the Map classes
- It, like all Map classes, HashMap exists in the java.util package
- A Map is an unordered group of key value pairs
- Keys must be unique (and immutable, ideally) whilst values can be duplicated
- Key value pairs may be added and removed
- Getting and setting values requires the specification of a key
- Use a HashMap when you need not process all elements but rather when you intend to access individual elements

HashMap

Essential Collections

```
// instantiation (type params = key type, value type)  
var populations = new HashMap<String, Integer>();
```

```
// adding/setting (value is overwritten if key exists)  
populations.put("London", 8982000);
```

```
// membership testing  
var londonIsPresent = populations.containsKey("London");
```

```
// sizing  
var numPairs = populations.size();
```

HashMap

Essential Collections

// getting

```
var londonPopulation = populations.get("London");
```

// removing

```
var removedValue = populations.remove("London");
```

// iterating (keys)

```
for (var city : populations.keySet()) { ... }
```

// iterating (values)

```
for (var population : populations.values()) { ... }
```