Arrays Class

Arrays Class

- `import java.util.Arrays;`
- Contains methods to manipulate arrays
- Syntax: <ClassName>.<methodName>(<parameter(s)>)
  - `Arrays.<methodName>(arrayName);`
  - Recap on similar ones: Math.sqrt(5), Character.toUpperCase('a')

Print an array

- Approach 1:
  - Print each element with a for loop
- Approach 2:
  - `Arrays.toString(arrayName)`

```java
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String[] args) {
        int[] arr = { 1, 2, 3 };
        System.out.println(Arrays.toString(arr));
    }
}
```

```
$ javac ArraysMethodDemo.java
$ java ArraysMethodDemo
[1, 2, 3]
```

Compare arrays (if two arrays are equal)

- Approach 1:
  - Compare size, if same, compare each element with a for loop
- Approach 2:
  - `Arrays.equals(array1, array2)`

```java
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String[] args) {
        int[] a1 = { 1, 2, 3 };
        int[] a2 = { 1, 2, 3 };
        System.out.println(Arrays.equals(a1, a2));
    }
}
```

```
$ javac ArraysMethodDemo.java
$ java ArraysMethodDemo
true
```

```java
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String[] args) {
        int[] a1 = { 1, 2, 3 };
        int[] a2 = { 1, 2, 3 };
        int[] a3 = a1;

        // == is used to compare referential equality
        System.out.println("a1 == a2 : " + (a1 == a2));   // false
        System.out.println("a1 == a3 : " + (a1 == a3));   // true
        System.out.println("a2 == a3 : " + (a2 == a3));   // false

        // By defualt, Object.equals() compares object memory addresses
        // Hence, it works the same as the == operator for Arrays
        System.out.println("a1.equals(a2) : " + a1.equals(a2));  // false
        System.out.println("a1.equals(a3) : " + a1.equals(a3));  // true
        System.out.println("a2.equals(a3) : " + a2.equals(a3));  // false

        // MUST use Arrays.equals(array1, array2) to compare arrays
        System.out.println("Arrays.equals(a1, a2) : " + Arrays.equals(a1, a2)); //true
        System.out.println("Arrays.equals(a1, a3) : " + Arrays.equals(a1, a3)); //true
        System.out.println("Arrays.equals(a2, a3) : " + Arrays.equals(a2, a3)); //true
    }
}
```

Resize an existing array

- Approach 1:
  - Construct another array, copy the elements from the old array to the new array with a for loop
- Approach 2:
  - `Arrays.copyOf(arrayName, newSize)`

```
$ javac ArraysMethodDemo.java
$ java ArraysMethodDemo
[1, 2, 3, 0, 0]
[1, 2]
[1, 2, 0, 0]
```

```
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String []args) {
        int[] arr = { 1, 2, 3 };

        // Expand the size to 5
        arr = Arrays.copyOf(arr, 5);
        System.out.println(Arrays.toString(arr));

        // Shrink the size to 2
        arr = Arrays.copyOf(arr, 2);
        System.out.println(Arrays.toString(arr));

        // Expand the size to 4
        arr = Arrays.copyOf(arr, 4);
        System.out.println(Arrays.toString(arr));
    }
}
```

Sort an array

`void sort(arrayName)`

- Sorts the array into ascending order

```
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String []args) {
        int[] arr = { 3, 5, 1, 4, 2 };

        Arrays.sort(arr);
        System.out.println(Arrays.toString(arr));

    }
}
```

```
$ javac ArraysMethodDemo.java
$ java ArraysMethodDemo
[1, 2, 3, 4, 5]
```

Search an element in an array

`int binarySearch(arrayName, value)`

- Returns the index of the given value in a **sorted** array if exists
- Returns a negative number if the value doesn't exist

```java
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String []args) {
        int[] arr = { 3, 5, 1, 4, 2 };
        Arrays.sort(arr);

        int loc0 = Arrays.binarySearch(arr, 0);
        System.out.println("0 is found at " + loc0);

        int loc2 = Arrays.binarySearch(arr, 2);
        System.out.println("2 is found at " + loc2);
    }
}
```

```
$ javac ArraysMethodDemo.java
$ java ArraysMethodDemo
0 is found at -1
2 is found at 1
```

Fill an array with a same value

`void fill(arrayName, value)`

- Sets every element in the array to the given value

```
import java.util.Arrays;

public class ArraysMethodDemo {
    public static void main(String []args) {
        int[] arr = { 3, 5, 1, 4, 2 };

        System.out.println("Before: " + Arrays.toString(arr));

        Arrays.fill(arr, 2);
        System.out.println("After: " + Arrays.toString(arr));
    }
}
```

```
$ javac ArraysMethodDemo.java
$ java ArraysMethodDemo
Before: [3, 5, 1, 4, 2]
After: [2, 2, 2, 2, 2]
```