Java traits

New Austrian Coding School

Programming - memo

The **reason** why we use objects is because of what they can do, their **traits**. We will **master** those traits to become masters of **data manipulation**.

String traits

- isEmpty()
- equals(string)
- equalsIgnoreCase(string)
- startsWith(string)
- contains(string)
- endsWith(string)
- toUpperCase()
- toLowerCase()
- trim()
- length()
- substring(start, end)
- replaceAll(target, replacement)
- split(separator)
- matches(regex)

Understanding the static keyword

Sometimes we want **something** to be done, **regardless** of who does it. The **static** keyword allows us to use methods **without** using objects.

Math traits

The Math class static methods

- Math.ceil(double)
- Math.floor(double)
- Math.abs(number)

Randomization

The Random object

- nextInt()
- nextInt(limit)
- nextDouble()
- nextBoolean()

Collection traits

The Collections class static methods

- Collections.reverse(collection)
- Collections.rotate(collection, shift)
- Collections.sort(collection)
- Collections.shuffle(collection)

Object traits

○ Every class **extends** from Object ○ Every class has **common** methods	○ toString() ○ equals(other) ○ compareTo(other)

Changing types

○ A Double **is not** an Integer ○ A Integer **is not** a String ○ What happens when we want to **transform** an Integer into a Double, or an Integer into a String? ○ What happens when we want to **transform** a String into an Integer?	○ new Double(integer) ○ number.toString() ○ Integer.valueOf(string)