Streams

Using Optional

What is an Optional?

- Optional is a container object that is used to contain values
- think of it as a box, which can be either empty or it can contain something
  - but the Optional object itself (the box) is never null
- there are methods to deal with optional values without explicit null checks
  - otherwise, we could get `NullPointerException` if not handled properly
- we have to import java.util.Optional

```java
// example: we want to calculate the average value of the numbers
// in the array which might be empty
public static Optional<Double> average (int... myNumbers) {
  if (myNumbers.length == 0)
    return Optional.empty();
  int sum = 0;
  for (int number : myNumbers)
    sum += number;
  return Optional.of((double) sum / myNumbers.length);
}

System.out.println(average(10, 20, 30));
System.out.println(average());
```

```
Optional[20.0]
Optional.empty
```

```
// sometimes we want to check if the Optional is empty
Optional<Double> myOptional = average(10, 20, 30);
if (myOptional.isPresent())  first we check if myOptional contains a value
  System.out.println(myOptional.get());  if so, get the value  ( => 20.0 )


// otherwise we could get the exception
Optional<Double> myEmptyOptional = average();
System.out.println(myEmptyOptional.get());
  => NoSuchElementException
```

```java
// good practice is to use empty() when value is null

Optional myOptional = (value == null) ? Optional.empty() : Optional.of(value);


// there is a factory method which takes care of this

Optional myNullableOptional = Optional.ofNullable(value);
```

if the value is null, Optional.empty is returned

```java
// run the method if Optional is not empty

Optional<Double> myOpt = average(10, 20, 30);

myOpt.ifPresent(System.out::println);
```

prints the value if the value is present, otherwise does nothing

Common Optional instance methods

Method	**When Optional is empty**	**When Optional Contains a value**
`get()`	Throws exception	Returns value
`ifPresent(Consumer c)`	Does nothing	Calls `Consumer` with value
`isPresent()`	Returns `false`	Returns `true`
`orElse(T other)`	Returns `other`	Returns value
`orElseGet(Supplier s)`	Returns result of `Supplier`	Returns value
`orElseThrow()`	`NoSuchElementException`	Returns value
`orElseThrow(Supplier s)`	Throws exception in Supplier	Returns value

```
Optional<Double> myOptional = average();

System.out.println(myOptional.orElse(Double.NaN));

  => NaN
```

the supplier must provide a double !!

```
System.out.println(myOptional.orElseGet(() -> Math.random()));

  => 0.4170449049884586   // generated by the supplier in the runtime!

System.out.println(myOptional.orElseThrow());

  => NoSuchElementException

System.out.println(myOptional.orElseThrow(() -> new IllegalStateException()));

  => IllegalStateException


// if myOptional wasn't empty, the value will be returned in all cases
```