Define `Unit`.

An abstract final class that extends `AnyVal`.

Its only instance is `()`.

What compromises did Scala make for Java interop?

- Static overloading of methods.
- Both traits and classes.
- Inclusion of null.
- Huge number of static types (e.g., existential types).

Give a structural type example for parameter that defines a `getName` function.

```
def printName(namedObj: {def.getName(): String}) {
  println(namedObj.getName())
}
```

Describe `Nothing`.

A final trait that extends `Any`.

Subtype of every `Any`.

There are no instances of this type at all.

Describe `None`.

A case object that extends `Option[Nothing]`.

This represents non-existent values.

Describe `Nil`.

A case object extending `List[Nothing]`.

It can stand in as an empty `List` of any type.

Describe `Null` and `null`.

Final trait that extends `AnyRef`.

Subtype of `AnyRef`.

Its only instance is the `null` reference.