

# Wrapping Up

Good design guidelines often recommend that we should prefer delegation over inheritance. Yet that was a hard sell in many OO languages, due to lack of direct support for delegation. Kotlin fixes that by providing facilities to delegate calls on objects, and access to both local variables and properties. Using the `by` keyword, you may delegate reads to any object that implements a `getValue()` method and writes to any object with `setValue()` method. Using these low-ceremony facilities, you may create your own custom delegates easily. You can also benefit from a few built-in standard delegates in the Kotlin standard library, like the Lazy delegate for example.

Being a multiparadigm language, Kotlin not only offers extensive OO programming features, as we've seen, it also provides phenomenal support for functional programming, as we'll see next.

---

In the next chapter, we'll explore functional style and lambda expressions.