

Wrapping Up

It's highly likely that you'll have a need to intermix Kotlin and Java code on your projects. In this chapter we've seen how easy it is to use Java from Kotlin, but the other way around requires some effort and planning.

You may use any Java library from your Kotlin applications and also use Kotlin libraries from within Java applications. You may also use both Java and Kotlin source files side by side in your projects. The Kotlin compiler has facilities to enable joint compilation of source files written in Java and Kotlin.

Calling into Java code from Kotlin is fairly straightforward and mostly works without any effort. You may use a pair of backticks to escape any keyword conflicts that may arise. Calling from Java to Kotlin requires planning and effort. The reason for this stems from the differences between Kotlin and Java; in particular, the enhanced features are available only in Kotlin. Using annotations that are part of the Kotlin standard library, you can customize the bytecode generated by the Kotlin compiler so you can more easily call it from Java.

Creating automated tests is a great way to make sure that integration efforts are successful. And tests can provide fast feedback that code continues to work as expected as it evolves.

In the next chapter, we'll take a look at writing unit tests for code written in Kotlin.