

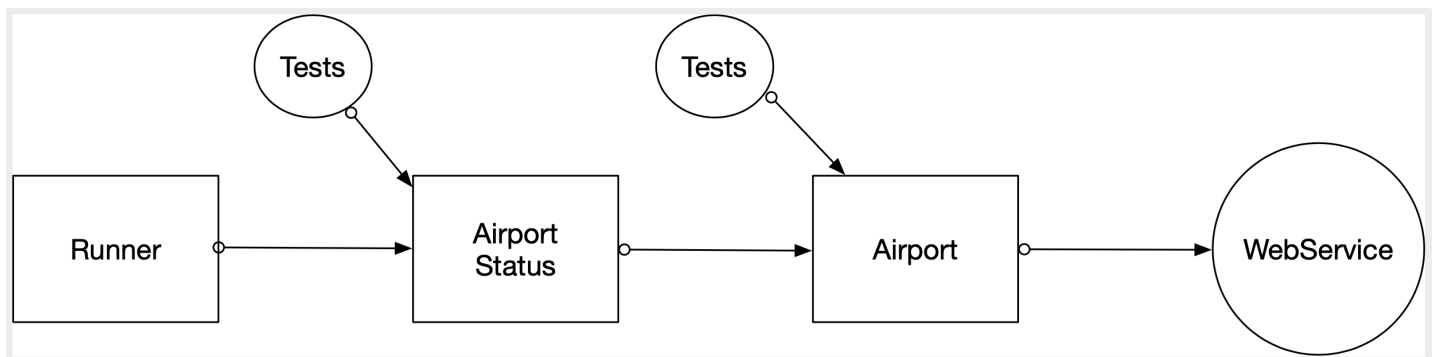
The Code Under Test

We'll cover the following ^

- Application design
- Testing approach

Application design

To practice writing automated tests for Kotlin code, we'll pick a sample application. We'll create an asynchronous program that will print, in sorted order of name, the status of different airports. The program will fetch the necessary data from a remote web service, like we saw in [Chapter 17, Asynchronous Programming](#). The overall design of the application that we'll create is shown in the following figure.



The `AirportApp.kt` file will hold the `main()` function to invoke a top-level function, written in the `AirportStatus.kt` file, within a `com.agiledeveloper.airportstatus` package. That function will use an `Airport` data class to hold the data for each airport of interest and to fetch the data from the remote web service.

The design will be implemented from right to left. The `WebService`, thankfully, already exists—we'll use the FAA website for airport information. We'll create, using tests, the `Airport` class first with methods to sort a list of airports, to get the data from the remote service, and to parse the JSON data. Then we'll implement, again with tests, the function in the `AirportStatus.kt` file. Finally, we'll write the `main()` function in `AirportApp.kt`.

Testing approach

We'll take the test-first approach—that is, we'll follow short cycles: write a small useful test first and then minimum code to make the test pass. At any time we'll be a maximum of one step away from code that passes all the tests.

Before we write any code, we have to make sure the necessary automated testing tool is set up properly. To test code written in Kotlin, you may use the traditional Java unit-testing tools: the good old JUnit 4 or the much newer JUnit 5. In this chapter, though, we'll use [KotlinTest](#), a unit testing tool targeted at unit testing Kotlin code. The benefit of using KotlinTest over JUnit is that the tests can follow the Kotlin idiomatic style. KotlinTest also provides facilities to create property-based tests and also data-driven tests.

The first step should be to set up a project with the necessary tools. Let's do that in the next lesson.