***Summary***

AOP is a powerful complement to object-oriented programming. With aspects, you can group application behavior that was once spread throughout your applications into reusable modules. You can then declare exactly where and how this behavior is applied. This reduces code duplication and lets your classes focus on their main functionality. Spring provides an AOP framework that lets you insert aspects around method executions. You’ve learned how to weave advice before, after, and around a method invocation, as well as to add custom behavior for handling exceptions.

You have several choices in how you can use aspects in your Spring applications. Wiring advice and pointcuts in Spring is much easier with the addition of @AspectJ annotation support and a simplified configuration schema.

Finally, there are times when Spring AOP isn’t enough, and you must turn to AspectJ for more powerful aspects. For those situations, we looked at how to use Spring to inject dependencies into AspectJ aspects.