The **Builder pattern** is a **good choice when designing classes whose constructors or static factories would have more than a handful of parameters**.

We've all at some point encountered a class with a list of constructors where each addition adds a new option parameter:

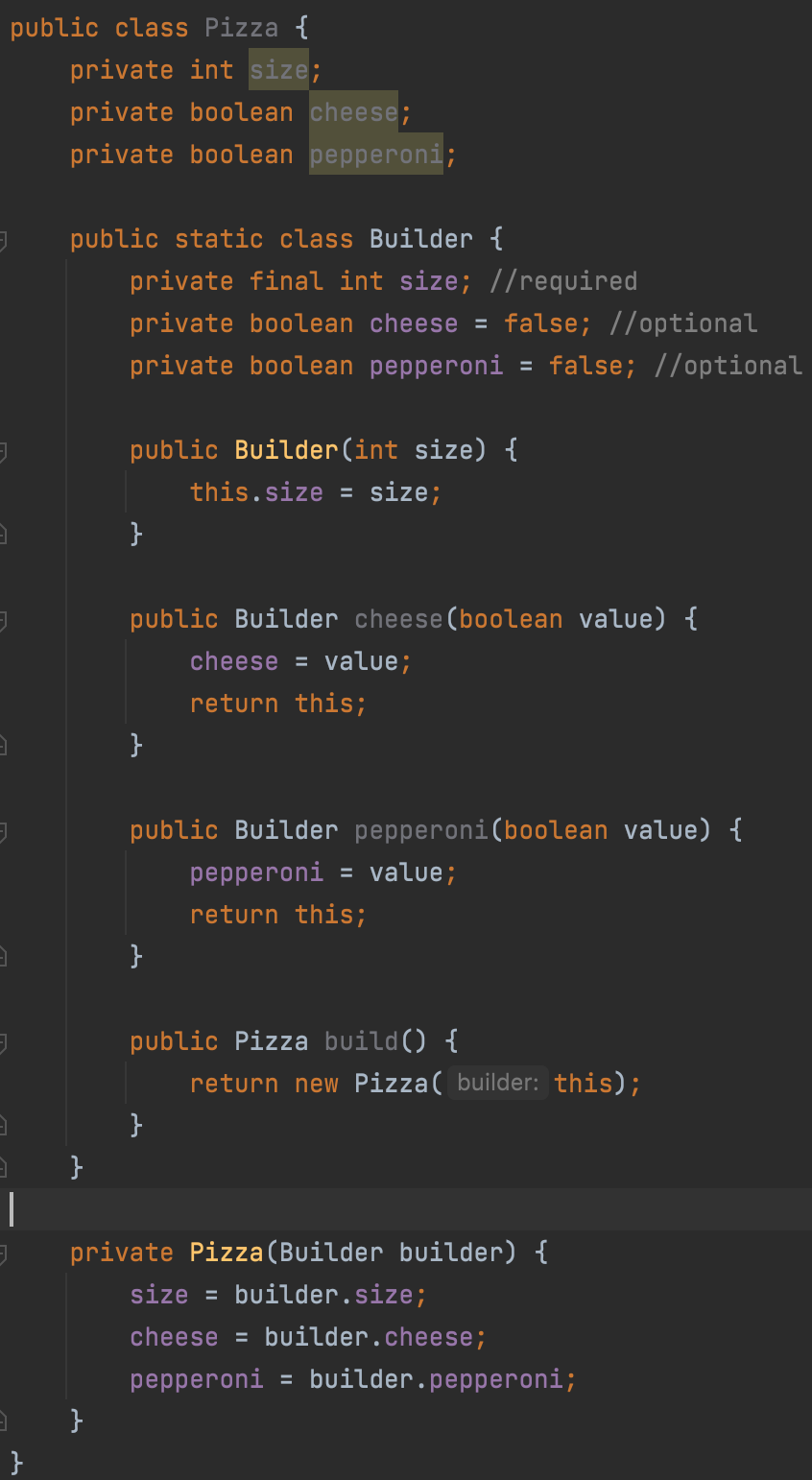
Pizza(int size) { ... }

Pizza(int size, boolean cheese) { ... }

Pizza(int size, boolean cheese, boolean pepperoni) { ... }

Pizza(int size, boolean cheese, boolean pepperoni, boolean bacon) { ... }

This is called the **Telescoping Constructor Pattern**. The problem with this pattern is that once constructors are 4 or 5 parameters long it **becomes difficult to remember the required order of the parameters** as well as what particular constructor you might want in a given situation.



Note that **Pizza is immutable** and that parameter values are all in a single location. Because the **Builder's setter methods return the Builder object they are able to be chained**.

