20.1 Introduction

You've used existing generic methods and classes in Chapters 7 and 16. In this chapter, you'll learn how to write your own.

It would be nice if we could write a single `sort` method to sort the elements in an `Integer` array, a `String` array or an array of any type that supports ordering (i.e., its elements can be compared). It would also be nice if we could write a single `Stack` class that could be used as a `Stack` of integers, a `Stack` of floating-point numbers, a `Stack` of `String`s or a `Stack` of any other type. It would be even nicer if we could detect type mismatches at *compile time*—known as **compile-time type safety**. For example, if a `Stack` should store only integers, an attempt to push a `String` onto that `Stack` should issue a *compilation* error. This chapter discusses **generics**—specifically **generic methods** and **generic classes**—which provide the means to create the type-safe general models mentioned above.