16.1 Introduction

In Section 7.16, we introduced the generic `ArrayList` collection—a dynamically resizable array-like data structure that stores references to objects of a type that you specify when you create the `ArrayList`. In this chapter, we continue our discussion of the Java **collections framework**, which contains many other *prebuilt* generic data-structures.

Some examples of collections are your favorite songs stored on your smartphone or media player, your contacts list, the cards you hold in a card game, the members of your favorite sports team and the courses you take at school.

We discuss the collections-framework interfaces that declare the capabilities of each collection type, various classes that implement these interfaces, methods that process collection objects, and **iterators** that "walk through" collections.

Java SE 8

8

After reading Chapter 17, Lambdas and Streams, you'll be able to reimplement many of Chapter 16's examples in a more concise and elegant manner, and in a way that makes them easier to parallelize to improve performance on today's multi-core systems. In Chapter 23, Concurrency, you’ll learn how to improve performance on multi-core systems using Java’s *concurrent collections* and *parallel stream* operations.

Java SE 9

9

Section 16.14 introduces Java SE 9’s new *convenience factory methods*, which help you create small immutable collections that cannot be modified once they’re created.