

16 Generic Collections

Objectives

In this chapter you'll:

- Learn what collections are.
- Use class `Arrays` for array manipulations.
- Learn the type-wrapper classes that enable programs to process primitive data values as objects.
- Understand the boxing and unboxing that occurs automatically between objects of the type-wrapper classes and their corresponding primitive types.
- Use prebuilt generic data structures from the collections framework.
- Use various algorithms of the `Collections` class to process collections.
- Use iterators to “walk through” a collection.
- Learn about synchronization and modifiability wrappers.
- Learn about Java SE 9’s new factory methods for creating small immutable `Lists`, `Sets` and `Maps`.

Outline

1. [16.1 Introduction](#)
2. [16.2 Collections Overview](#)

- 3. [16.3 Type-Wrapper Classes](#)
- 4. [16.4 Autoboxing and Auto-Unboxing](#)
- 5. [16.5 Interface Collection and Class Collections](#)
- 6. [16.6 Lists](#)
 - 1. [16.6.1 ArrayList and Iterator](#)
 - 2. [16.6.2 LinkedList](#)
- 7. [16.7 Collections Methods](#)
 - 1. [16.7.1 Method sort](#)
 - 2. [16.7.2 Method shuffle](#)
 - 3. [16.7.3 Methods reverse, fill, copy, max and min](#)
 - 4. [16.7.4 Method binarySearch](#)
 - 5. [16.7.5 Methods addAll, frequency and disjoint](#)
- 8. [16.8 Class PriorityQueue and Interface Queue](#)
- 9. [16.9 Sets](#)
- 10. [16.10 Maps](#)
- 11. [16.11 Synchronized Collections](#)
- 12. [16.12 Unmodifiable Collections](#)
- 13. [16.13 Abstract Implementations](#)
- 14. [16.14 Java SE 9: Convenience Factory Methods for Immutable Collections](#)
- 15. [16.15 Wrap-Up](#)
 - 1. [Summary](#)
 - 2. [Self-Review Exercises](#)
 - 3. [Answers to Self-Review Exercises](#)
 - 4. [Exercises](#)