Java Collections Framework

Presented by developerWorks, your source for great tutorials

ibm.com/developerWorks

Table of Contents

If you're viewing this document online, you can click any of the topics below to link directly to that section.

Section 1. Tutorial tips

Should I take this tutorial?

This tutorial takes you on an extended tour of the Java Collections Framework. The tutorial starts with a few simple programming examples for beginners and experts alike, to get started with the Collections Framework quickly. The tutorial continues with a discussion of sets and maps, their properties, and how their mathematical definition differs from the `Set`, `Map`, and `Collection` definitions within the Collections Framework. A section on the history of Java Collections Framework clears up some of the confusion around the proliferation of set- and map-like classes. This tutorial includes a thorough presentation of all the interfaces and their implementation classes in the Collections Framework. The tutorial explores the algorithm support for the collections, as well as working with collections in a thread-safe and read-only manner. In addition, the tutorial includes a discussion of using a subset of the Collections Framework with JDK 1.1. The tutorial concludes with an introduction of JGL, a widely used algorithm and data structure library from ObjectSpace that predates the Java Collections Framework.

Concepts

At the end of this tutorial you will know the following:

* The mathematical meaning of set, map, and collection
* The six key interfaces of the Collections Framework

Objectives

By the end of this tutorial, you will know how to do the following:

* Use the concrete collection implementations
* Apply sorting and searching through collections
* Use read-only and thread-safe collections

Contact

jGuru has been dedicated to promoting the growth of the Java technology community through evangelism, education, and software since 1995. You can find out more about their activities, including their huge collection of FAQs at *jGuru.com* . To send feedback to jGuru about this tutorial, send mail to *producer@jguru.com* .

Course author: John Zukowski does strategic Java consulting for *JZ Ventures, Inc.* . His latest book is *"Java Collections"* (Apress, May 2001).

Section 2. Collections Framework

Introduction

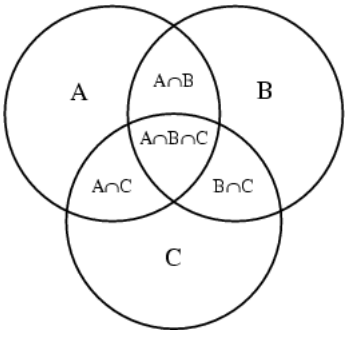
This tutorial takes you on an extended tour of the Collections Framework, first introduced with the Java 2 platform, Standard Edition, version 1.2. The Collections Framework provides a well-designed set of interfaces and classes for storing and manipulating groups of data as a single unit, a *collection*. The framework provides a convenient API to many of the abstract data types familiar from computer science data structure curriculum: maps, sets, lists, trees, arrays, hashtables, and other collections. Because of their object-oriented design, the Java classes in the Collections Framework encapsulate both the data structures and the algorithms associated with these abstractions. The framework provides a standard programming interface to many of the most common abstractions, without burdening the programmer with too many procedures and interfaces. The operations supported by the collections framework nevertheless permit the programmer to easily define higher-level data abstractions, such as stacks, queues, and thread-safe collections.

One thing worth noting early on is that while the framework is included with the Java 2 platform, a subset form is available for use with Java 1.1 run-time environments. The framework subset is discussed in Working with the Collections Framework support in JDK 1.1 on page 33.

Before diving into the Collections Framework, it helps to understand some of the terminology and set theory involved when working with the framework.

Mathematical background

In common usage, a *collection* is the same as the intuitive, mathematical concept of a *set*. A set is just a group of unique items, meaning that the group contains no duplicates. The Collections Framework, in fact, includes a `Set` interface, and a number of concrete `Set` classes. But the formal notion of a set predates Java technology by a century, when the British mathematician George Boole defined it in formal logic. Most people learned some set theory in elementary school when introduced to "set intersection" and "set union" through the familiar Venn Diagrams:



Some real-world examples of sets include the following:

* The set of uppercase letters 'A' through 'Z'
* The set of non-negative integers {0, 1, 2 ...}