

15 Files, Input/Output Streams, NIO and XML Serialization

Objectives

In this chapter you'll:

- Create, read, write and update files.
- Retrieve information about files and directories using features of the NIO.2 APIs.
- Learn the differences between text files and binary files.
- Use class `Formatter` to output text to a file.
- Use class `Scanner` to input text from a file.
- Use sequential file processing to develop a real-world credit-inquiry program.
- Write objects to and read objects from a file using XML serialization and the JAXB (Java Architecture for XML Binding) APIs.
- Use a `JFileChooser` dialog to allow users to select files or directories on disk.
- Optionally use `java.io` interfaces and classes to perform byte-based and character-based input and output.

Outline

1. [15.1 Introduction](#)
2. [15.2 Files and Streams](#)
3. [15.3 Using NIO Classes and Interfaces to Get File and Directory Information](#)
4. [15.4 Sequential Text Files](#)
 1. [15.4.1 Creating a Sequential Text File](#)
 2. [15.4.2 Reading Data from a Sequential Text File](#)
 3. [15.4.3 Case Study: A Credit-Inquiry Program](#)
 4. [15.4.4 Updating Sequential Files](#)
5. [15.5 XML Serialization](#)
 1. [15.5.1 Creating a Sequential File Using XML Serialization](#)
 2. [15.5.2 Reading and Deserializing Data from a Sequential File](#)
6. [15.6 FileChooser and DirectoryChooser dialogs](#)
7. [15.7 \(Optional\) Additional java.io Classes](#)
 1. [15.7.1 Interfaces and Classes for Byte-Based Input and Output](#)
 2. [15.7.2 Interfaces and Classes for Character-Based Input and Output](#)
8. [15.8 Wrap-Up](#)
 1. [Summary](#)
 2. [Self-Review Exercises](#)
 3. [Answers to Self-Review Exercises](#)
 4. [Exercises](#)

5. Making a Difference