「一入 Java 深似海」系列课程 - 第八期

第一节：Java I/O 流（Streams）

小马哥 @mercyblitz

「一入 Java 深似海」系列课程

- 讲师信息

小马哥，Java 劝退师，Apache 和 Spring Cloud 等知名开源架构成员。

- 微博：@mercyblitz
- GitHub：https://github.com/mercyblitz
- 课程详情：https://segmentfault.com/n/1330000017785588
- 课件资源：https://github.com/mercyblitz/segmentfault-lessons/

segmentfault 讲堂

主要议题

- Java I/O 字节流（Byte Streams）
- Java I/O 字符流（Character Streams）
- Java I/O 流操作

Java I/O Byte Streams

Java I/O Byte Streams

- 基本概念

Programs use byte streams to perform input and output of 8-bit bytes. All byte stream classes are descended from InputStream and OutputStream.There are many byte stream classes:

- 直接实现
  - 文件 - FileInputStream/FileOutputStream
  - 数组 - ByteArrayInputStream/ByteArrayOutputStream
- 装饰器模式实现

Java I/O Byte Streams

- 装饰器模式实现
  - 缓冲 - BufferedInputStream/BufferedOutputStream
  - 数据 - DataInputStream/DataOutputStream
  - 过滤器 - FilterInputStream/FilterOutputStream
  - 对象 - ObjectInputStream/ObjectOutputStream
  - 管道 - PipedOutputStream/PipedOutputStream

Java I/O Character Streams

Java I/O Character Streams

- 基本概念

The Java platform stores character values using Unicode conventions. Character stream I/O automatically translates this internal format to and from the local character set. In Western locales, the local character set is usually an 8-bit superset of ASCII.For most applications, I/O with character streams is no more complicated than I/O with byte streams. Input and output done with stream classes automatically translates to and from the local character set. A program that uses character streams in place of byte streams automatically adapts to the local character set and is ready for internationalization — all without extra effort by the programmer.

If internationalization isn't a priority, you can simply use the character stream classes without paying much attention to character set issues. Later, if internationalization becomes a priority, your program can be adapted without extensive recoding.

Java I/O Character Streams

- 直接实现
  - 文件 - FileInputStream/FileOutputStream
  - 字符串 - StringReader/StringWriter
- 装饰器模式实现
  - 缓冲 - BufferedReader/BufferedWriter
  - 管道 - PipedReader/PipedWriter

Java I/O 流操作

Java I/O 流操作

- 扫描（Scanning）
- 格式化（Formatting）
- 命令行交互（Command-Line）
- 数据流操作（Data Streams）
- 对象序列化操作（Object Streams）

下期预告

Java 文件系统

谢谢观看