

ストリームでの入出力処理の基本

Android開発のためのJavaSE再入門（6）

JavaGoogleAndroid

印刷用を表示

🔖 ブックマーク

🐦 ツイート

9

🔗 シェアする

📖 B! 24

📖 Pocket

46

WINGSプロジェクト 高江 賢[著] / 山田 祥寛[監修]

2011/12/14 14:00

ダウンロード サンプルファイル (45.4 KB)

目次

ファイル入出力の基本

ファイルに対して、ある処理の結果を書き込んだり、何らかのデータを読みとったりすることは、実際のアプリケーションにおいては、非常によく用いられる処理です。

ここでは、このようなファイル入出力のプログラムを、ストリームを用いて行う処理の基本例として、解説することにします。

なおファイル処理は、プログラムを実行するハードウェアに依存する部分があります。まずはWindows上で動作確認できるサンプルで一般的な説明を行い、記事の後半で、Andoridでのプログラム例を説明します。

ファイルからの入力

最初にファイルの入力処理です。ファイルでは、文字でもバイナリでもあつかえますが、今回は文字ストリームを用いたテキストファイルの例を見ていきましょう。

次のサンプルコードは、sample1.txtというテキストファイルを1行ずつ読み込み、標準出力に書きだす例です。

【リスト1】 Sample1クラス

```
try {
    File f = new File("sample1.txt"); // (1)
    FileReader fr = new FileReader(f); // (2)
    BufferedReader br = new BufferedReader(fr); // (3)

    String line;
    try {
        // 1行ずつファイルの最後まで読み込む
        while ((line = br.readLine()) != null) { // (4)
            System.out.println(line);
        }
    } finally {
        br.close(); // (5)
    }
} catch (IOException e) {
    // ファイルがない等のエラーがあった場合の処理
}
```

1. Fileオブジェクトの生成

java.io.Fileクラスは、ファイルやディレクトリを操作するためのクラスです。ファイルやディレクトリの生成や存在確認、アクセス権や名称の調査など、さまざまなメソッドが用意されています。ここでは、ファイル名を引数としてFileオブジェクトを生成しています。

2. FileReaderオブジェクトの生成

java.io.FileReaderクラスは、実際にファイルをオープンして文字を読み込むストリームクラスです。この例では、コンストラクタにFileオブジェクトを指定していますが、直接ファイル名を渡すことも可能です。その場合は、Fileクラスは不要です。

3. BufferedReaderオブジェクトの生成

java.io.BufferedReaderクラスは、FileReaderオブジェクトのデータを一時的に保持するクラスで、プログラムとの橋渡しとなるストリームクラスです。つまりファイルからのデータは、2つのストリームをたどるということです。ストリームオブジェクトは、少なくともひとつは必要ですが、このように連結することもできます。APIには、いろいろな処理に応じたストリームクラスが用意されています。なお、この例のファイルの読み込み処理は、BufferedReaderクラスを経由しないでも実現可能です。ただファイルやネットワーク通信のように、アクセスに少し時間がかかるデータに対しては、一時的にデータを蓄える（バッファ）ストリームオブジェクトを利用するのが通常です。バッファを介することで、物理的なアクセス回数が減り、プログラムの処理が速くなるからです。

4. BufferdReaderのreadLine（またはread）メソッドで読み込む

BufferedReaderクラスのreadLineメソッドは、改行文字（`¥n`）、復帰文字（`¥r`）、復帰改行文字（`¥r¥n`）のいずれかまでを1行とするテキストを読み込み、文字列として返します。結果の文字列には、終端の文字は含まれません。また、ストリームの終わりを検出すると、nullを返します。もうひとつの読み込みメソッドであるreadメソッドは、1文字単位でデータを読み込む場合に利用します。

5. 各オブジェクトの終了処理

Javaでは、オブジェクトの破棄はガベージコレクションにより自動で行われます。明示的に終了処理を記述する必要がないため、ストリームのクローズ処理も忘れがちになります。もちろんストリームオブジェクトが破棄されるときにクローズされるのですが、ストリームに関しては、使用時のみオープンして、使用後は速やかに閉じるのが基本です。このサンプルでは、BufferedReaderオブジェクトのcloseメソッドのみ呼んでいます。BufferedReaderオブジェクトは、FileReaderオブジェクトを引数として生成しています。つまり、連結されたストリームオブジェクトは、数珠つなぎになっているのです。最後のストリームオブジェクトをクローズすることで、連結されたストリームすべてがクローズされ、その過程でファイルも解放されることになります。なお、closeメソッドの呼び出しは、finallyブロックに記述するようにしましょう。これは、ファイルの読み込み中に例外が発生した場合でも、クローズ処理が漏れないようにするためです。

文字コードの指定

このようにFileReaderクラスは、お手軽に利用できるクラスですが、残念な面もあるクラスです。それは、明示的に文字コードを指定して読み込めないことです。

FileReaderクラスでは、Java仮想マシンのデフォルト・エンコーディングにしか対応していないため、日本語などで文字化けが発生する場合があります。文字コードを指定してテキストファイルを読み込みたい場合には、FileReaderクラスの代わりに、バイト単位のストリームクラス、FileInputStreamクラスとInputStreamReaderクラスを組み合わせます。

例えば、Windowsで用いられる文字コード、Windows-31Jのテキストファイルであれば、次のようになります。InputStreamReaderの第2引数に、文字コードを指定します。

```
InputStreamReader isr = new InputStreamReader(
    new FileInputStream(f), "Windows-31J"); // (2)
BufferedReader br = new BufferedReader(isr); // (3)
```

ファイルへの出力

テキストファイルに出力する基本的な手順は、入力処理と同じような手順になります。Fileオブジェクトを生成した後、FileWriter（またはOutputStreamWriterとFileOutputStreamのペア）とBufferedWriterのオブジェクトを生成します。そして、BufferedWriterクラスのwriteメソッドを呼び出して、データを書きだします。

具体的な例は、次のAndoridで動作するサンプルで紹介することにしましよう。

次ページ» Androidのファイル入出力処理

メンバーメニュー オプション

顧客コード10020

漢字 龍崎太郎

氏名 カナ ブドウタロウ

性別

budou.taro@grapecity.com

ふいがな取得や入力検証？簡単にできるよ？

» 快適で正確な操作を促す入力フォームを開発

INPUTMAN

Special Contents

「攻めのインフラ」で顧客の要件に寄り添うインフラの専門家集団、grasysで働く魅力とは

SREは運用チームだけの問題？ 開発者のメリットをGoogle×スリーシェイクがブラクティスとともに解説！

厳選！キャリアインタビュー

よりITが事業に貢献できるところで活躍したい——不動産テックを支えるSREエンジニアの働き方

人気ランキング

今日 月間

- Googleがプログラミング言語「Go 1.19」を公開、メモリモデルに対する変更など
- ローコードで管理画面を開発できるサービス「Dashcomb」が登場
- エクセルソフト、テスト自動化ツール「TestComplete」を日本で販売開始、SmartBearのGUIテスト自動化ツール
- 手軽に高速にWebサイトを公開！Webホスティングサービス「Vercel」の概要と使い方
- データ分析を容易にするKibanaハンズオン〜データを適切な形で可視化しよう〜
- Vue.jsでWebページをつくる際の肝！「コンポーネント」をTypeScriptで活用しよう
- TypeScriptを効率的に独習しよう！無料で学べる「TypeScript Deep Dive」日本語版の翻訳者が学習法を解説
- 「Docker Desktop 4.9」が公開、Windows版がVMware ESXiなどに新たに対応
- HashiCorp、使い慣れたプログラミング言語でのインフラ構築を可能にする「CDK for Terraform」を一般提供
- Vue.jsの世代交代が到来！Vue 3デフォルト時代の「Vue.js開発新常識」

新着

記事 ニュース

データ分析を容易にするKibanaハンズオン〜データを適切な形で可視化しよう〜

VercelにNext.jsアプリケーションをデプロイする〜Vercel上でNext.jsを動かす利点も紹介

Vue.jsでWebページをつくる際の肝！「コンポーネント」をTypeScriptで活用しよう

「ディレクトリとパス」〜マンガでプログラミング用語解説

事業をスケールさせるエンジニアへの転身で、技術のコモディティ化に打ち勝つ【デブスト2021】

📖 新着記事一覧を見る

Pick Up Links

シスコDevNetとは？ プログラマブルな最新のネットワークを学べるコミュニティ

未来を創るエンジニアたちが見る世界 ヤフー特集／各領域の多彩なエンジニアが語り合う未来像【PR】

Claris FileMakerとは？ ビジネス課題をITで解決する、ローコード開発ツールを大解剖

広告掲載・イベント協賛 問い合わせ 広告出稿、イベント出展についてのお問い合わせはこちら

Flash Builder 4.6でAndroid／iOSアプリ開発！ Android／iOS固有の機能を使ったアプリもFlash Builderで！

価値を生む開発に集中しつづける現場インタビュー〜クラウド最前線

アプリケーション開発を支援するコンポーネントジン ComponentZine アプリケーション開発の生産性と品質の向上を支援する

インフラ技術のトップランナーたちと行く、開発者のためのSRE探求

IRIS Data Platform 特集 RPA、AI時代のデータ活用、データプラットフォームを考える

UNIXコマンド辞典ショートカット

- ファイル操作
- システム管理
- ネットワーク管理
- 印刷処理
- インストール
- テキスト処理
- ジョブ管理
- デバイス処理
- 圧縮・解凍
- UNIX基本講座