Android開発のためのJava SE再入門

Google

印刷用を表示

Android開発のためのJavaSE再入門(6)

Android

ダウンロード ↓ サンプルファイル (45.4 KB)

WINGSプロジェクト 高江 賢[著] / 山田 祥寛[監修]

Androidのファイル入出力処理

Pocket < 46

2011/12/14 14:00

メンバーメニュー

Qサイト内検索

検索

オプション

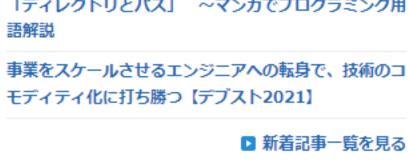


# PR Special Contents SREは運用チームだけの問題? 開発者

#### 厳選!キャリアインタビュー PR 決済業界を牽引するGMOペイメントゲ ートウェイ CTOに訊く、多様な決済シ



- 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
- 新着 記事 ニュース



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

```
Android/iOS固有の機能
 Android / iOS
               を使ったアプリもFlash
  アプリ開発!
               Builderで!
               広告出稿、イベント出展に
広告掲載・イベント協賛
               ついてのお問い合わせはこ
```



ファイル操作

▶ インストール

ネットワーク管理

がプラクティスとともに解説!

ステムを創るエンジニア組織と開発の

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

データ分析を容易にするKibanaハンズオン~デ **一夕を適切な形で可視化しよう~** 

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

インフラ構築を可能にする「CDK for

Terraform」を一般提供

HashiCorp、使い慣れたプログラミング言語での

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~

Vue.jsでWebページをつくる際の肝!「コンポーネン

Vercel上でNext.jsを動かす利点も紹介

ト」をTypeScriptで活用しよう

新着記事一覧を見る Pick Up Links

### 問い合わせ ちら

ビジネス課題をITで解決

ルを大解剖

クラウド最前線

▶ システム管理

▶ テキスト処理

▶ 印刷処理

する、ローコード開発ツー

価値を生む開発に集中しつ

づける現場インタビュー~

RPA、AI時代のデータ活



価値を生む開発に集中しつづける

▶ ジョブ管理 ▶ デバイス処理 ▶ 圧縮・解凍 ▶ UNIX基本講座

のメリットをGoogle×スリーシェイク 「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

実態とは? 人気ランキング 今日 月間

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、

Vue.jsでWebページをつくる際の肝!「コンポ

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」

「ディレクトリとパス」 ~マンガでプログラミング用

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

インフラ技術の インフラ技術のトップラン トップランナーたちと行く、 ナーたちと行く、開発者の ためのSRE探求 Flash Builder 4.6で

RIS Data Platform 用、データプラットフォー ムを考える 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]

UNIXコマンド辞典ショートカット

ストリームでの入出力処理の基本

У ツイート ⟨ 9 (介 シェアする 5

# 【New】デブサミ2022夏 デブサミ2022 デブスト2021 デブサミウーマン

ホーム ニュース 記事▼ 特集ページ一覧

# Developers Online 主催イベント マ その他 マ

#### サンプルコードの紹介の前に、少しAndroidでのファイル入出力処理全般の話をしておきます。 Androidでも、ファイル入出力処理は、これまで説明した手順と大差はありません。ただ注意すべ き点(主にハードウェアに依存する部分)もあります。 入出力ファイルの位置

OutputStreamWriter osw = new OutputStreamWriter(

- // 改行の出力

BufferedWriter bw = new BufferedWriter(osw);

Androidでは、任意の場所にあるファイルを、自由に操作することは許されていません。端末の 内部にあるストレージなら、アプリケーションから入出力できるのは、/data/data/「パッケージ 名]/files/以下のファイルと決められています。そのため、パス名を指定してストリームオブジェク

目次

#### トを生成する場合、このパスを指定しないとエラーになってしまいます。

専用メソッド

Androidでは、ファイルのストリームを取得するための専用メソッドとして、openFileInput、 openFileOutputというふたつのメソッドがContextクラスに用意されています。これらのメソッド では、規定のパス名の指定が不要になります(指定できない)。また戻り値として、 FileInputStream/FileOutputStreamオブジェクトを返します。

#### 次のサンプルは、openFileOutputメソッドを使って、ファイルにテキストを書きだす例です。 try {

try {

finally {

bw.write("abc");

bw.newLine();

されていることが確認できます。

¥ 5554:AVD-1

try {

[リスト2] main.xml

</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>

android:orientation="vertical"

ンポーネントを追加する形をとっています。

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

createTable(f); // ファイル作成

// ベースのレイアウト取得

String f = "sample.txt";

} catch (IOException e) {

android:id="@+id/parent"

android:layout\_width="fill\_parent" android:layout\_height="fill\_parent"

export PATH=/data/local/bin:\$PATH

\$ cd /data/data/codezine.androidjava.chap6/files \$ cat sample.txt

ファイル出力

bw.close(); catch (IOException e) {

openFileOutput("sample.txt" Context.MODE\_WORLD\_READABLE));

```
なお、openFileOutputメソッドの第2引数はファイルの操作モードです。次の定数が指定でき、
or(|)で区切ることで複数指定も可能です。
操作モードに指定可能な定数
定数
                      概要
                      追記モードで開く
MODE_APPEND
                      作成したアプリケーションのみアクセス可能にする
MODE_PRIVATE
MODE_WORLD_READABLE
                      他アプリケーションの読み込み可にする
                      他アプリケーションの書き込み可にする
MODE_WORLD_WRITEABLE
 Androidエミュレータでファイルが作成されていることを確かめるには、インストールされてい
```

A S D F G H J K L 🖔 & Z X C V B N M

ALT SYM @

**a** 10:23



[リスト3] HelloAndroid.java public class HelloAndroid extends Activity { public void onCreate(Bundle savedInstanceState) {

LinearLayout parent = (LinearLayout) findViewById(R.id.parent);

サンプルソースは、次のようになります。画面表示は、ベースとなるレイアウトに、TextViewコ

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</p>

// ファイルに九九表を出力する public void createTable(String f) { try { PrintWriter pw = new PrintWriter(new BufferedWriter( new OutputStreamWriter(openFileOutput(f, Context.MODE\_WORLD\_READABLE)))); try { for (int i = 1; i < 10; i++) { for (int j = 1; j < 10; j++) { pw.printf("%3d ", i \* j); pw.printf("\mathbb{n}"); } finally { pw.close(); } catch (IOException e) { まずcreateTableメソッドで、ファイルに九九表を出力しています。さきほどのPrintWriterクラ スを用いて、数値を書き込んでいます。 ファイルの読み込みは、openFileInputメソッドで入力ストリームを取得し、 InputStreamReaderとBufferedReaderクラスを使って、1行ずつ処理しています。取得した1行 は、それぞれテキストビューに設定して(setTextメソッド)、レイアウトに追加(addViewメソッ ド)しています。なお、ここでは、フォントを等幅にするために、setTypefaceメソッドで、フォ ントファミリーにTypeface.MONOSPACEの指定をしています。ちなみにこれ以外では、 Typeface.SERIF、Typeface.SANS\_SERIFが指定できます。 実行結果は次のようになります。もちろんファイルにも、まったく同様の内容が出力されていま す。 図2 実行結果 O G X 5554:AVD-1 The control of th HelloAndroid

# 最後に

参考資料

今回は、ストリームを用いたファイル入出力の基本的な処理を説明しました。次回は、基本API

の中で、日付の処理など、まだ説明していなかったAPIをとりあげることにします。

3. Android Developers

<<前^

るTerminalEmulatorアプリ([Dev Tools] にある)を起動します。次のように、ファイルが作成 図1 TerminalEmulatorでcatコマンドの表示

try { BufferedReader br = new BufferedReader(new InputStreamReader( openFileInput(f))); try { String line; while ((line = br.readLine()) != null) { TextView tv = new TextView(this); tv.setTypeface(Typeface.MONOSPACE); // 等幅フォントの指定 tv.setText(line); // ビューの追加 parent.addView(tv); } finally { br.close();

10 15 20 25 30 35 40 45 12 16 20 24 28 32 36 6 10 12 14 16 18 9 12 15 18 21 24 27 21 28 35 42 49 56 63 4 5 6 7 8 42 48 54 56 64 72 72

5 6 7 8 9

V B N M

G H

S D

C

16

24

32 40 48

14

6

8

## 『Javaポケットリファレンス』 WINGSプロジェクト 高江賢 著、山田祥寛 監修、技術評論

社、2011年3月

2. Java Platform, Standard Edition 6 API 仕様