

JavaプログラミングのTip集

第2回

2007年5月5日
アイディアス



推奨されない API

Javaバージョンによりコンパイル後、警告の生じることがあります

```
>javac NetServerforTest.java
```

注:NetServerforTest.java は推奨されない API を使用またはオーバーライドしています。

注:詳細については、-Xlint:deprecation オプションを指定して再コンパイルしてください。

```
>javac -Xlint NetServerforTest.java
NetServerforTest.java:32: 警告:[deprecation] java.io.DataInputStream の readLine
() は推奨されません。
    while( (line = dis.readLine()) != null ){
                ^
警告 1 個
```

↑
disオブジェクトの定義を変更する



推奨されない DataInputStreamのreadLine()

```
Socket sock = null; //sockオブジェクトの宣言
InputStream is = null; //isオブジェクトの宣言
DataInputStream dis = null; //disオブジェクトの宣言
String line; //行データ
```

```
sock = servsock.accept(); //クライアントの接続要求待ち
is = sock.getInputStream();
dis = new DataInputStream(is);
while( (line = dis.readLine()) != null ){ } 推奨されない使い方
    System.out.println("data..." + line);
}
```



DataInputStreamから BufferedReaderへ置き換え

```
Socket sock = null; //sockオブジェクトの宣言
InputStream is = null; //isオブジェクトの宣言
InputStreamReader ir = null; // irオブジェクトの宣言
BufferedReader br = null; //brオブジェクトの宣言
String line; //行データ
```

```
sock = servsock.accept(); //クライアントの接続要求待ち
is = sock.getInputStream();
ir = new InputStreamReader(is);
br = new BufferedReader(ir);
while( (line = br.readLine()) != null ){ } 対策済みの使い方
    System.out.println("data..." + line);
}
```



バイト配列をバイトストリームに書き込み

```
byte[] data = null; //受信データ
ByteArrayOutputStream ba = null; //baオブジェクトの宣言
```

```
data = line.getBytes(); ← 行データ
```

```
ba = new ByteArrayOutputStream();
ba.write(data);
```



ServerSocketの使い方

```
ServerSocket servsock; //servsockオブジェクトの宣言
```

```
try{
    servsock = new ServerSocket(Integer.parseInt(args[0])); 第一引数を使用
    while(true){
        try{
            sock = servsock.accept(); //クライアントの接続要求待ち
            ← サーバー側処理を記述
        }catch(NullPointerException e){ //ヌルエラー出力処理
            System.err.println("null trouble");
        }catch(IOException e){ //ioエラー出力処理
            System.err.println("io trouble");
        }
        sock.close(); //ソケット接続終了
    }
}catch(IOException e){ //標準エラー出力処理
    System.err.println("io trouble");
}
```

