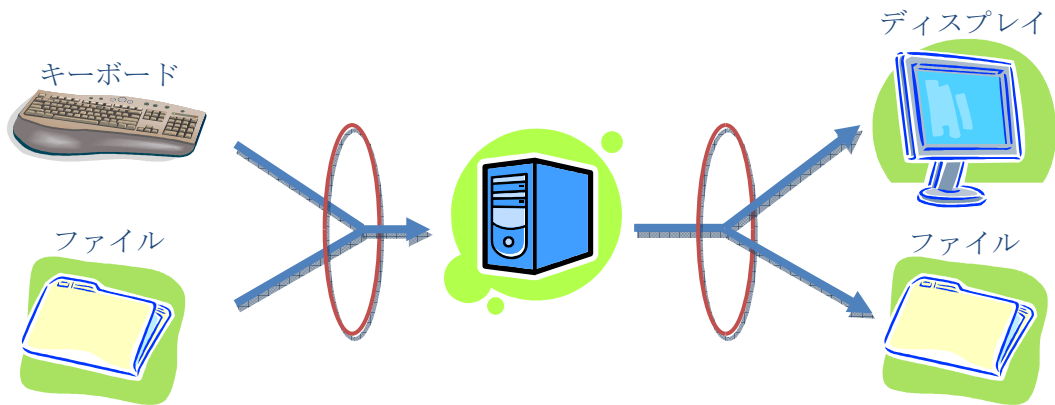


ストリーム

ストリームとは 入力、または出力の流れをもつデータの連鎖
 ストリームは異なる機器からのデータを統一的に扱うようにする考え方

文字ストリーム 文字ベースの入出力をサポートする
 - Unicode (Java 内部) とローカルコード (Java 外部) の変換が行われ、プログラマはその違いを意識しなくてもよい

バイトストリーム バイナリデータの入出力をサポートする
 - 画像データや音楽データなどのバイナリデータをありのままに読み込み、また保存できる



I/O クラス ストリームを定義する
 OS 依存の細かな情報を隠しプログラマに様々な入出力を提供する
 ※I/O クラスは `import java.io.*;` で利用できる

標準入出力を提供するクラス

クラス	クラス変数	操作対象	機能
System	in	キーボード	標準入力
System	out	ディスプレイ	標準出力
System	err	ディスプレイ	標準エラー出力※

※リダイレクト時も常にディスプレイに表示

ファイル入出力を提供するクラス

クラス	操作対象	機能
FileReader	ファイル	ファイル入力 ・文字単位の入力
FileWriter	ファイル	ファイル出力

- ・文字単位の出力

バッファ機能を提供するクラス

クラス	機能
BufferedReader	入力にバッファ機能を付加 ・ readLine()メソッドが含まれ、 1行読み込みが可能になる
BufferedWriter	出力にバッファ機能を付加 ・ newLine()メソッドが含まれ、 改行の出力が可能になる

便利なサービスを提供するクラス

クラス	機能
PrintWriter	書式付き出力メソッドを追加 ・ print()や println()メソッドが利用できる

一般的な文字ストリーム (入力)

キーボード	<pre> InputStreamReader isr; isr = new InputStreamReader(System.in); BufferedReader br; br = new BufferedReader(isr); String str = br.readLine(); </pre>	<p>(1) 標準入力 (バイトストリーム) を文字ストリームへ結合</p> <p>(2) バッファ機能を追加</p> <p>(注)クローズしない System.in や System.out、System.err は実行開始時にシステムがオブジェクトを予め生成して利用可能としているため</p>
ファイル	<pre> FileReader fr = new FileReader("ファイル名"); BufferedReader br = new BufferedReader(fr); String str = br.readLine(); br.close(); </pre>	<p>(1) ファイルに結びついたオブジェクトを生成</p> <p>(2) バッファ機能を追加</p> <p>(3) ファイルをクローズし、ファイルに関連付いたリソースを解放する</p>

一般的な文字ストリーム (出力)

ディスプレイ `System.out.println()`; 又は、

```
OutputStreamWriter osr;  
osr = new OutputStreamWriter( System.out );  
BufferedWriter bw = new BufferedWriter( osr );  
PrintWriter pw = new PrintWriter( bw );  
.....  
pw.println( );  
.....
```

(1) 標準出力 (バイトストリーム) を文字ストリームへ結合
(2) バッファ機能を追加
(3) `print()`や `println()`など書式付き出力メソッドを追加
(注)クローズしない

ファイル `FileWriter fw = new FileWriter("ファイル名");`
`BufferedWriter bw = new BufferedWriter(fw);`
`PrintWriter pw = new PrintWriter(bw);`
.....
`pw.println();`
.....
`pw.close();`

(1) ファイルに結びついたオブジェクトを生成
(2) バッファ機能を追加
(3) `print()`や `println()`など書式付き出力メソッドを追加
(4) ファイルをクローズ

《ファイルの追加書込みと上書き書込み》

```
FileWriter("ファイル名", true) → 追加書込み  
FileWriter("ファイル名", false) → 上書き書込み  
FileWriter("ファイル名") → 上書き書込み
```

コマンドライン引数

プログラムを実行する際にそのプログラムに与える値 (文字列) の列。パラメータともいう。

値の渡し方: パラメータなし `>java クラスファイル`
 パラメータあり `>java クラスファイル パラメータ1 パラメータ2`

値の受け方: `public static void main(String[] args){`
 `// args.length → パラメータの数 0 ~`
 `// args[i] → i 番目のパラメータの文字列`
 `}`