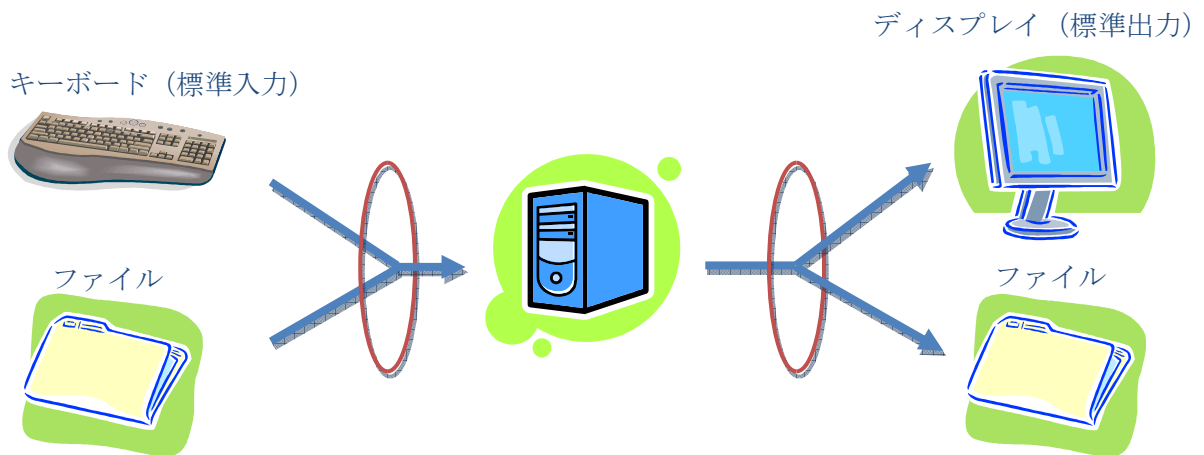


ストリーム

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



ストリームと I/O クラス

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

標準入力/標準出力を管理するクラスとオブジェクト

クラス	クラス変数	操作対象	機能	ストリーム
System	System.in ^{*1}	キーボード	標準入力	バイト
System	System.out ^{*1}	ディスプレイ	標準出力	バイト
System	System.err ^{*1}	ディスプレイ	標準エラー出力 ^{*2}	バイト

^{*1} System.in や System.out、System.err は実行開始時にシステムがそれぞれの操作対象に結び付いたオブジェクトを生成し代入して利用可能にする

^{*2} リダイレクト時も常にディスプレイに表示

各ストリーム間の変換をするクラス

クラス	機能	ストリーム
InputStreamReader	バイトストリーム（入力）を 文字ストリーム（入力）に変換	文字
OutputStreamWriter	バイトストリーム（出力）を 文字ストリーム（出力）に変換	文字

入力ファイル/出力ファイルを管理するクラス

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

バッファ機能を提供するクラス

クラス	機能	ストリーム
BufferedReader	文字ストリーム（入力）に バッファ機能を付加 ・ <code>readLine()</code> メソッドが含まれ、 1行読み込みが可能になる	文字
BufferedWriter	文字ストリーム（出力）出力に バッファ機能を付加 ・ <code>newLine()</code> メソッドが含まれ、 改行の出力が可能になる	文字

便利なサービスを提供するクラス

クラス	機能	ストリーム
PrintWriter	文字ストリーム（出力）に 書式付き出力機能を追加 ・ <code>print()</code> や <code>println()</code> メソッドが利用できる	文字

一般的な文字ストリーム (入力)

```
キーボード // (1) バイトストリーム (標準入力) を文字ストリームに変換
            InputStreamReader isr = new InputStreamReader( System.in );
            // (2) 文字ストリームにバッファ機能を追加
            BufferedReader br = new BufferedReader( isr );
            :
            String str = br.readLine();
            :
            // (注) System.in はシステムが準備したものであるため、クローズしない

ファイル // (1) ファイルに結び付いたオブジェクトを生成
            FileReader fr = new FileReader("ファイル名");
            // (2) 文字ストリームにバッファ機能を追加
            BufferedReader br = new BufferedReader( fr );
            :
            String str = br.readLine();
            :
            // (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
            br.close();
```

一般的な文字ストリーム (出力)

```
ディスプレイ System.out.println(); 又は、
            // (1) バイトストリーム (標準出力) を文字ストリームに変換
            OutputStreamWriter osr = new OutputStreamWriter( System.out );
            // (2) 文字ストリームにバッファ機能を追加
            BufferedWriter bw = new BufferedWriter( osr );
            // (3) さらに文字ストリームに print()や println()など書式付き出力機能を追加
            PrintWriter pw = new PrintWriter( bw );
            :
            pw.println();
            :
            // (注) System.out はシステムが準備したものであるため、クローズしない

ファイル // (1) ファイルに結び付いたオブジェクトを生成
            FileWriter fw = new FileWriter("ファイル名");
            // (2) 文字ストリームにバッファ機能を追加
            BufferedWriter bw = new BufferedWriter( fw );
            // (3) さらに文字ストリームに print()や println()など書式付き出力機能を追加
            PrintWriter pw = new PrintWriter( bw );
            :
            pw.println();
            :
            // (4) ファイルをクローズ
            pw.close();
```

《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`
 `}`