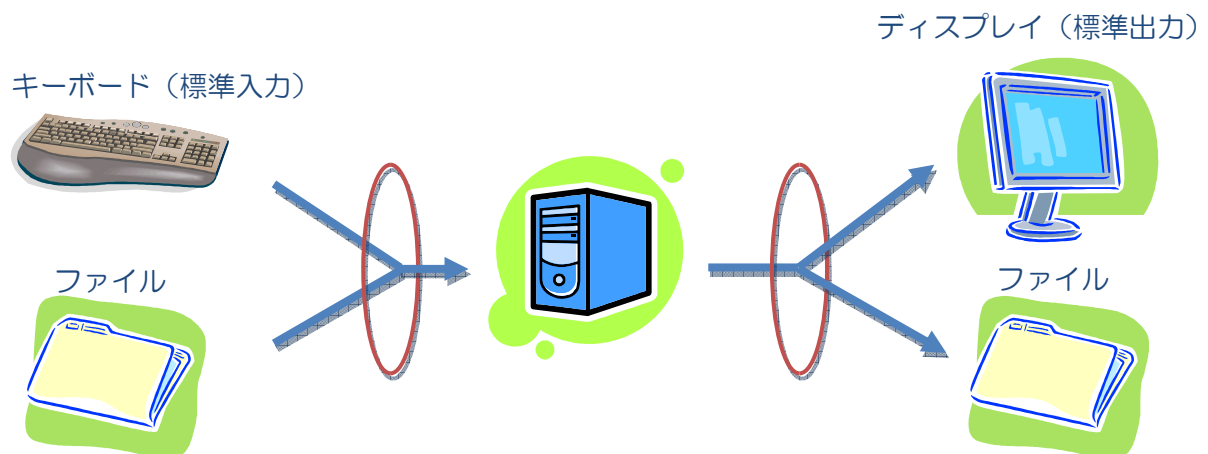


ストリーム

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です
ストリームは異なる機器からのデータを統一的に扱う考え方です

文字ストリーム 文字ベースの入出力をサポートします
- 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済みます

バイトストリーム バイナリデータの入出力をサポートします
- 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、また保存できます



文字ストリームとI/Oクラス

I/Oクラスとは ストリームを定義し、構築します
OS 依存の細かな情報を隠しプログラマに様々な入出力機能を提供します
※I/Oクラスは `import java.io.*;` で利用できます

ストリーム構築
のイメージ

I/Oクラスの
オブジェクト

標準入力 (キーボード) / 標準出力 (ディスプレイ) を表わすクラス

クラス	クラス変数	操作対象	要求ストリーム	提供ストリーム
System	InputStream in ^{*1}	標準入力	-	バイト (入力)
System	PrintStream out ^{*1}	標準出力	-	バイト (出力)

^{*1} System.in と System.out 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利用可能にします

各ストリーム間の変換をするクラス

クラス	機能	要求ストリーム	提供ストリーム
InputStreamReader	バイト (入力) を文字 (入力) にストリーム変換します	バイト (入力)	文字 (入力)
OutputStreamWriter	バイト (出力) を文字 (出力) にストリーム変換します	バイト (出力)	文字 (出力)

入カファイル/出カファイルを管理するクラス

クラス	操作対象	要求ストリーム	提供ストリーム
FileReader	入カファイル ・文字単位の入カ	-	文字 (入力)
FileWriter	出カファイル ・文字単位の出カ	-	文字 (出力)

バッファ機能を提供するクラス

クラス	機能	要求ストリーム	提供ストリーム
BufferedReader	文字 (入力) ストリームに バッファ機能を付加します ・readLine()メソッドを含み 1行読み込みが可能になります	文字 (入力)	文字 (入力)
BufferedWriter	文字 (出力) ストリームに バッファ機能を付加します ・newLine()メソッドを含み 改行の出カが可能になります	文字 (出力)	文字 (出力)

便利な出カサービスを提供するクラス

クラス	機能	要求ストリーム	提供ストリーム
PrintWriter	文字 (出力) ストリームに書式 付き出カ機能を追加します ・print()やprintln()メソッド を含み書式付出力ができます	文字 (出力)	文字 (出力)



一般的な文字 (入力) ストリームの構築

標準入力 // (1) バイト (入力) ストリーム (標準入力) を文字 (入力) ストリームに変換
InputStreamReader isr = new InputStreamReader(System.in);
// (2) 文字 (入力) 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) 文字 (入力) ストリーム (入力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 (入力) 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 (出力) ストリームの構築

標準出力 System.out.println(); 又は、
// (1) バイト (出力) ストリーム (標準出力) を文字 (出力) ストリームに変換
OutputStreamWriter osr = new OutputStreamWriter(System.out);
// (2) 文字 (出力) ストリームにバッファ機能を追加
BufferedWriter bw = new BufferedWriter(osr);
// (3) 文字 (出力) 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) 文字 (出力) ストリーム (出力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 (出力) ストリームにバッファ機能を追加
BufferedWriter bw = new BufferedWriter(fw);
// (3) 文字 (出力) 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

コマンドライン引数とは プログラムを実行する際にそのプログラムに与える文字列のリスト（パラメータともいいます）です

パラメータの渡し方：
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 . . .

args[0] args[1]

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