Ｊａｖａプログラミング Ⅱ

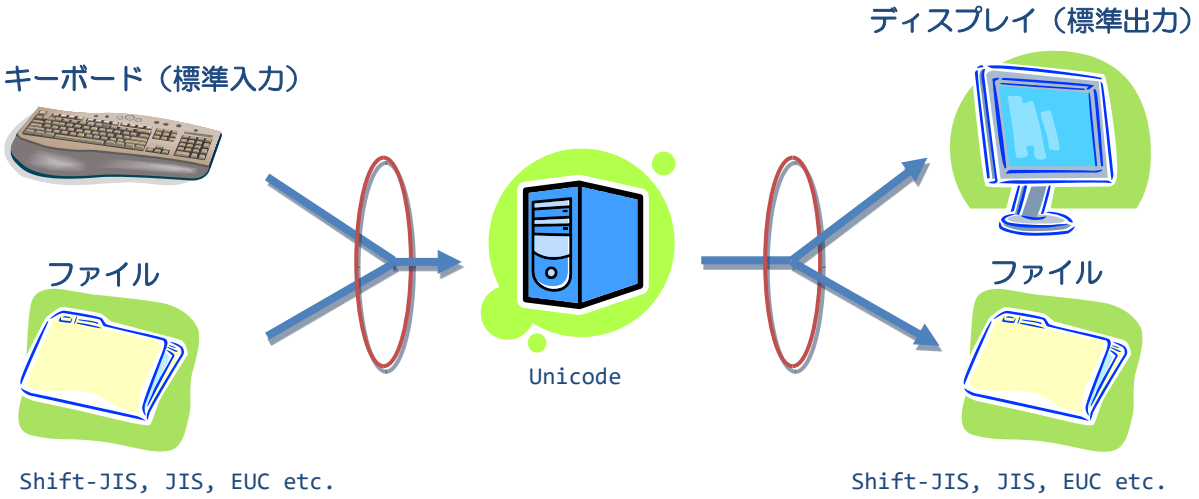
１０回目　ファイルへの出力

§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すみます

□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

□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	要求ストリーム	提供ストリーム
FileReader	入力ファイル ・文字単位の入力	-	文字（入力）
FileWriter	出力ファイル ・文字単位の出力	-	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力が可能になります	文字（出力）	文字（出力）

□便利な出力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力機能を追加します ・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ません
```

■ファイル
```
// (1) 文字(入力)ストリーム（入力ファイルに結び付くオブジェクト）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 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ません
```

■ファイル
```
// (1) 文字(出力)ストリーム（出力ファイルに結び付くオブジェクト）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 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書込み》

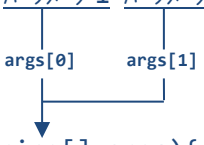
```
FileWriter("ﾌｧｲﾙ名",true)  → 追加書込み
FileWriter("ﾌｧｲﾙ名",false) → 上書書込み
FileWriter("ﾌｧｲﾙ名")       → 上書書込み
```

§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
クラスファイルには○○○.class の○○○部分を指定します

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 ・・・



■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 番目のパラメータです
   //                 args[0]が最初のパラメータです
}
```