

Javaプログラミング実習

23. ストリーム入出力

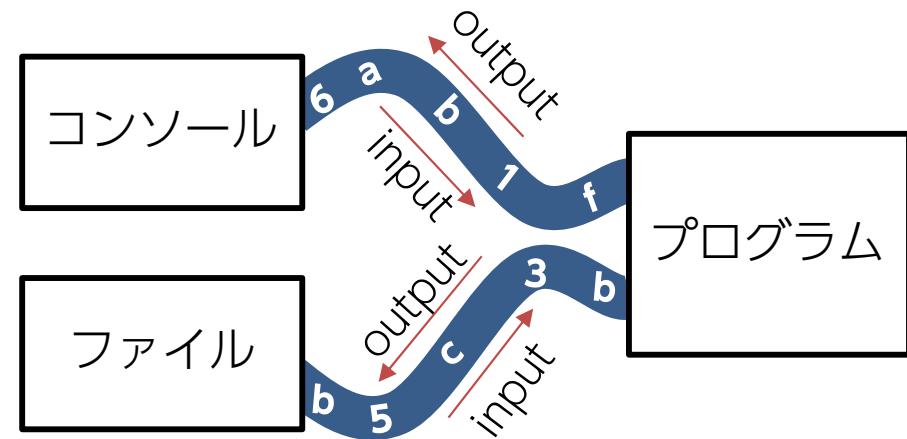
株式会社ジードライブ

今回学ぶこと

- ストリームの基礎
- 標準入出力の利用方法
- ファイルからのデータの読み出し方法
- ファイルへのデータの書き込み方法
- Fileクラスの概要と使い方

ストリームとは

- ここでのストリームは「データの流れ」といった意味合いを持つ
 - Javaではコンソールやファイルに対してのデータの入出力を一律に**ストリーム**として扱う
- ストリーム用のクラスは `java.io` パッケージに含まれる
 - `OutputStream`
 - `InputStream`
 - `PrintStream`
 - `FileOutputStream`
 - `FileInputStream` など



標準入出力

- コンソールへの出力を**標準出力**と呼ぶ
 - `System.out.println(...)` は標準出力
- コンソールを通じた入力を**標準入力**と呼ぶ
 - コンソールを通じたキーボード入力など
- Javaプログラムでは、起動時に標準入出力のストリームが自動的に生成される
 - 標準出力ストリーム： `System.out`
 - 標準入力ストリーム： `System.in`

コンソールへの出力

- OutputStream の一種であるPrintStreamを利用する

```
public class OutputTest {  
    public static void main(String[] args) {  
        PrintStream ps = System.out;  
        ps.println("Hello, Java!");  
        ps.close();  
    }  
}
```

ストリームは最後にclose()で閉じる

コンソール表示

Hello, Java!

実際には、あえてこのように書く必要はない(ファイル出力との比較のための記述)

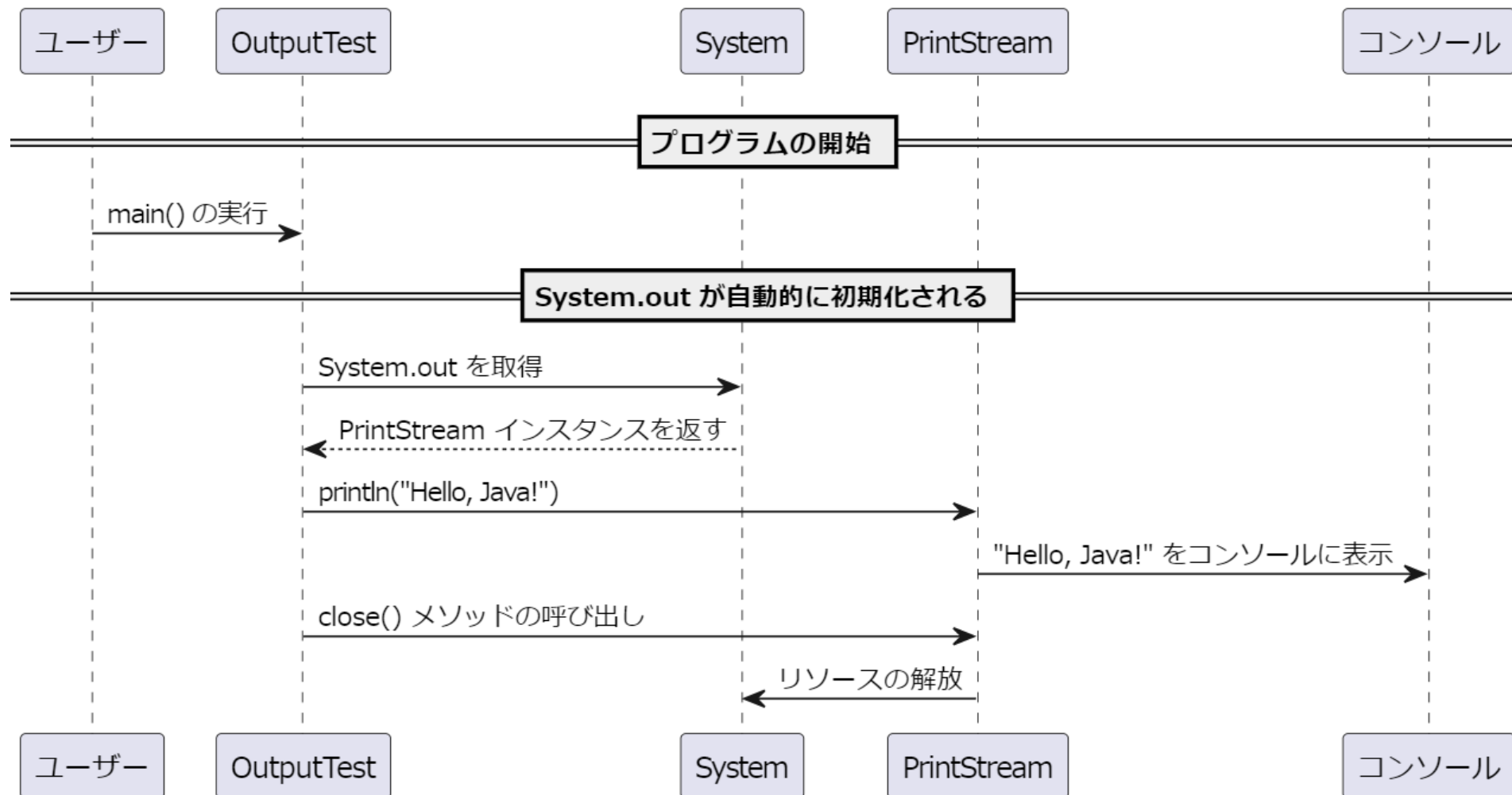
※ 標準の入出力のストリームに関してはプログラムの終了時に自動的に閉じられる

Systemクラス : <https://docs.oracle.com/javase/jp/21/docs/api/java.base/java/lang/System.html>

PrintStreamクラス : <https://docs.oracle.com/javase/jp/21/docs/api/java.base/java/io/PrintStream.html>

コンソールへの出力

前頁コードのシーケンス図



コンソールからの入力

- InputStreamを利用する

「例外処理」の教材で解説

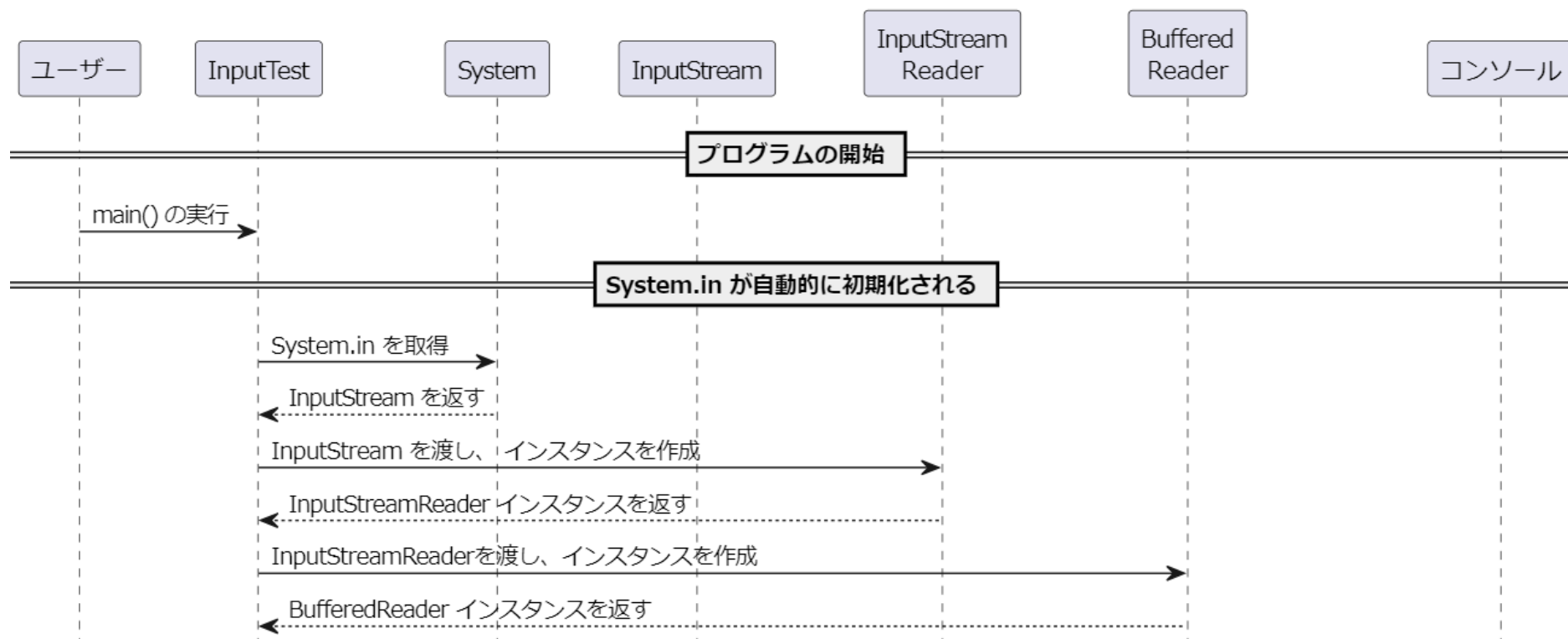
```
public class InputTest {  
    public static void main(String[] args) throws IOException {  
        InputStream is = System.in;  
        InputStreamReader isr = new InputStreamReader(is);  
        BufferedReader br = new BufferedReader(isr);  
        String text = br.readLine(); // 入力の読み取り  
        System.out.println("テキスト：" + text);  
        br.close();  
    }  
}
```

コンソールに Hello と入力し、
Enterを押した場合

Hello ←
テキスト : Hello

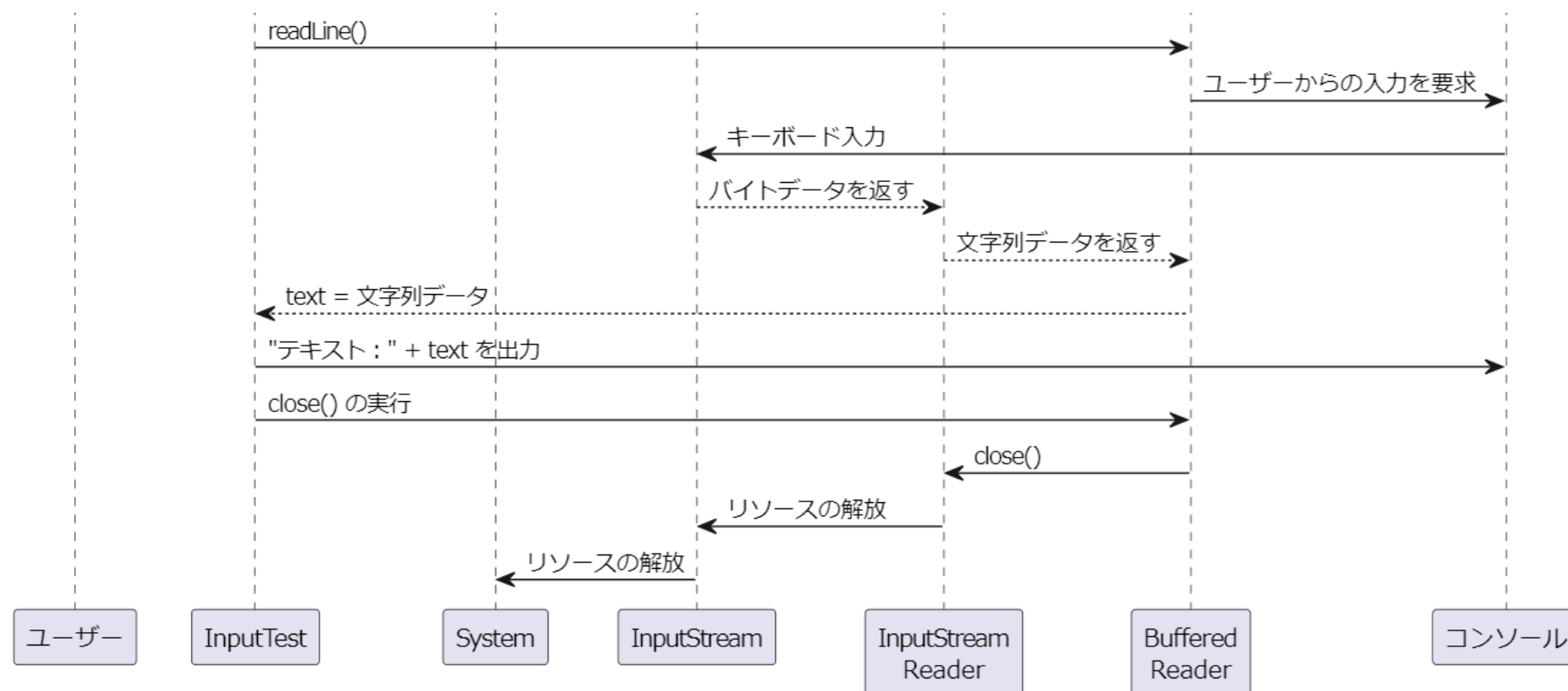
コンソールからの入力

前頁コードのシーケンス図



コンソールからの入力

前頁の続き



練習

- 練習23-1
- 練習23-2

ファイルへの出力

- 標準出力(コンソールへの出力) とは、
PrintStreamの取得元が異なる
 - 他の部分は同様の記述になる

```
public class FileOutputTest {  
    public static void main(String[] args) throws IOException {  
        PrintStream ps = new PrintStream("out.txt");  
        ps.println("こんにちは！");  
        ps.println("元気ですか？");  
        ps.close();  
    }  
}
```

コンソールへの出力
と同じメソッド

ファイルからの入力

- 標準入力(コンソールからの入力) とは、InputStreamの取得元が異なる
 - 他の部分は同様の記述になる

```
public class FileInputTest {  
    public static void main(String[] args) throws IOException {  
        InputStream is = new FileInputStream("data.txt");  
        InputStreamReader isr = new InputStreamReader(is);  
        BufferedReader br = new BufferedReader(isr);  
        String text = br.readLine();  
        System.out.println("テキスト：" + text);  
        br.close();  
    }  
}
```

標準入力と
同じ処理

ファイルからの入力

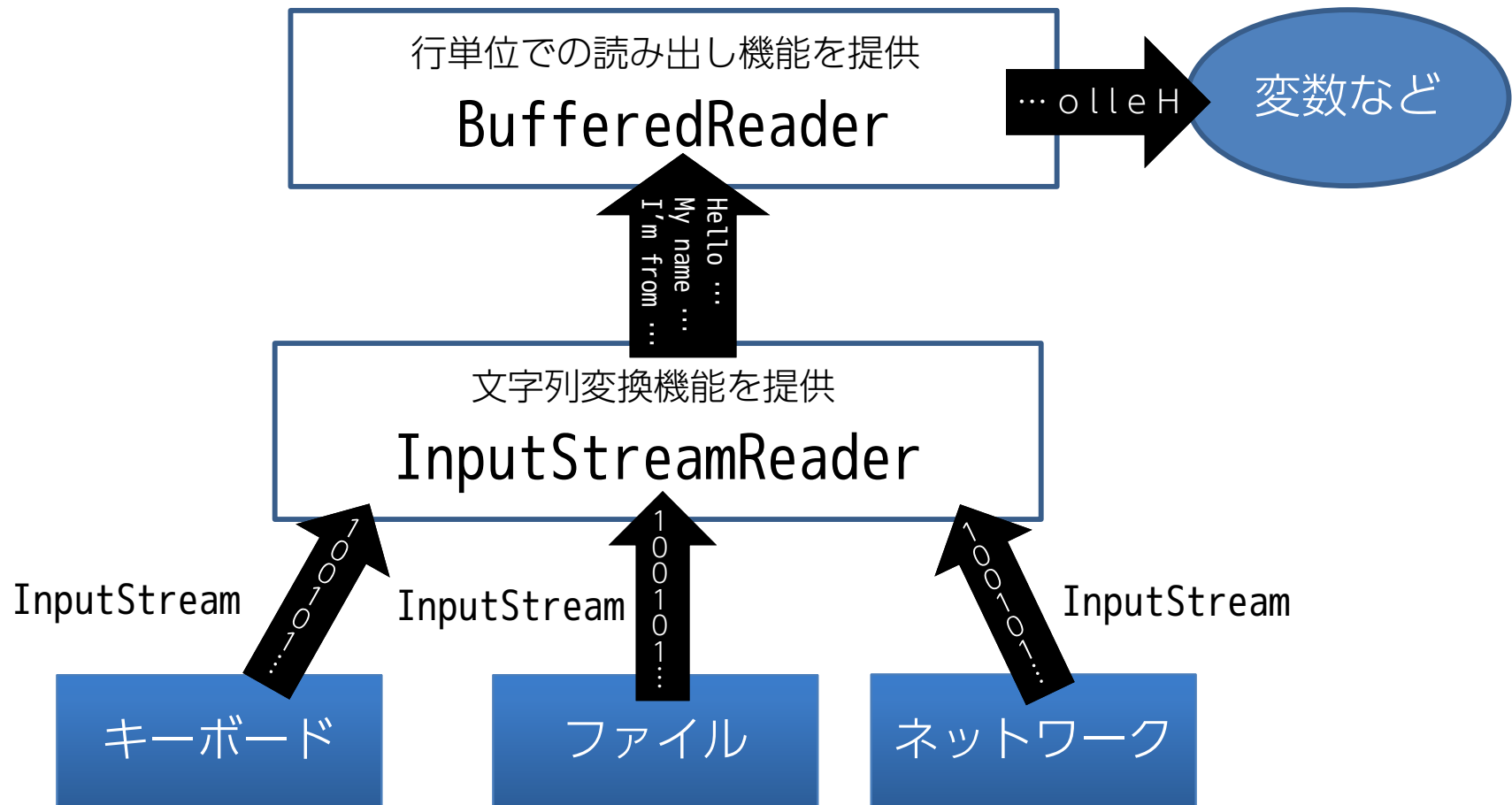
記述例：テキストファイルから全行を取得して標準出力する

```
public static void main(String[] args) throws IOException {  
    InputStream is = new FileInputStream("data.txt");  
    InputStreamReader isr = new InputStreamReader(is);  
    BufferedReader br = new BufferedReader(isr);  
    String line;  
    while(line = br.readLine() != null) {  
        System.out.println(line);  
    }  
    br.close();  
}
```

一行ずつ読み込み、
読み込む行がなくなるとnullを返す

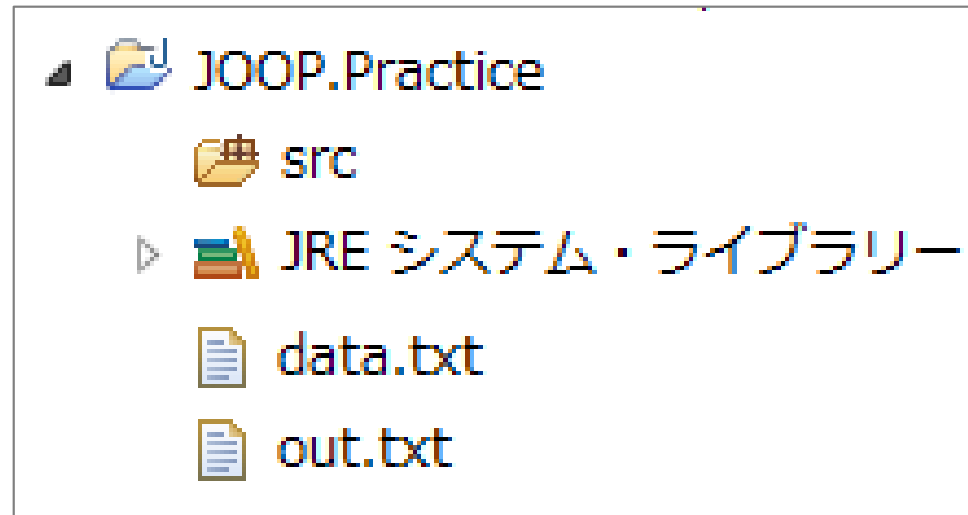
入カストリーム利用のイメージ

- InputStreamReaderを介することで、入力元の種類に関係なく同じメソッドでデータを読み出せる



ファイルの配置場所

- `new PrintStream()` や `new FileInputStream()` の引数としてファイル名のみを指定する場合 (ディレクトリを特に指定しない場合)、ファイルはプロジェクト直下に配置する



練習

- 練習23-3
- 練習23-4

Fileクラス

- ファイル/フォルダを操作する手段を提供するクラス
 - Fileという名前だが、フォルダも扱うことができる
 - 入出力関連のクラスではFileオブジェクトを引数とするものが多い
- Fileクラスを使ってできること
 - ファイル/フォルダの作成・削除・存在確認
 - ファイル/フォルダのパス、名前の取得
 - フォルダ内のファイル一覧の取得

Fileオブジェクトの生成

- 書式

```
File file = new File("ファイルのパス名");
```

例：絶対パス名による指定

```
File file = new File("C:/Users/zd1A00/Desktop/test01/test.txt");
```

例：相対パス名による指定

```
File file = new File("test.txt");
```

Eclipseで作業する場合、相対パス名はプロジェクトフォルダの直下からの相対的な位置となる

Fileクラスのメソッド

- 主なメソッド

メソッド	説明
<code>boolean createNewFile()</code>	新しいファイルを生成する（ファイルが存在しない場合のみ）
<code>boolean exists()</code>	ファイルもしくはフォルダが存在するかどうか判定する
<code>String getAbsolutePath()</code>	ファイルもしくはフォルダの絶対パス名を取得する
<code>String getParent()</code>	ファイルが格納されているフォルダの絶対パス名を取得する
<code>boolean isFile()</code>	ファイルかどうか判定する
<code>long length()</code>	ファイルのサイズを取得する
<code>File[] listFiles()</code>	フォルダ内のファイルを配列で取得する

Fileクラス：<https://docs.oracle.com/javase/jp/17/docs/api/java.base/java/io/File.html>

Fileクラスの使用例

例：デスクトップのmyfolder内のファイル名一覧を表示する

```
File dir = new File("C:/Users/zd1A00/Desktop/myfolder");  
for (File f : dir.listFiles()) {  
    if (f.isFile()) {  
        System.out.println(f.getName());  
    }  
}
```

コンソール表示例

```
info.text  
flower.png  
bird.jpg  
参考資料.pdf  
...
```

練習

- 練習23-5
- 練習23-6