

# 独習Javaゼミ

---

11.4 PrintWriterクラス

11.5 バイトストリーム

07/06/22

鈴木慧

# 11.4 PrintWriter クラス

---

## □ PrintWriter クラスとは

Writer を拡張したクラス

int、float、char などの基本データ型およびオブジェクトと  
等価の文字列を表示する

### PrintWriter コンストラクタ

PrintWriter(OutputStream 変数名)

PrintWriter(OutputStream 変数名, boolean 変数)

PrintWriter(Writer 変数名)

PrintWriter(Writer 変数名, boolean 変数)

# PrintWriter クラスの例

```
import java.io.*;

class PWDemo {
    public static void main(String args[]) {
        try {
            PrintWriter pw = new PrintWriter
                (System.out);
            pw.println(true);
            pw.println('A');
            pw.println(4000L);
            pw.println(45.67f);
            pw.println(45.67);
            pw.println("Hello");
            pw.println( new Integer("99") );
```

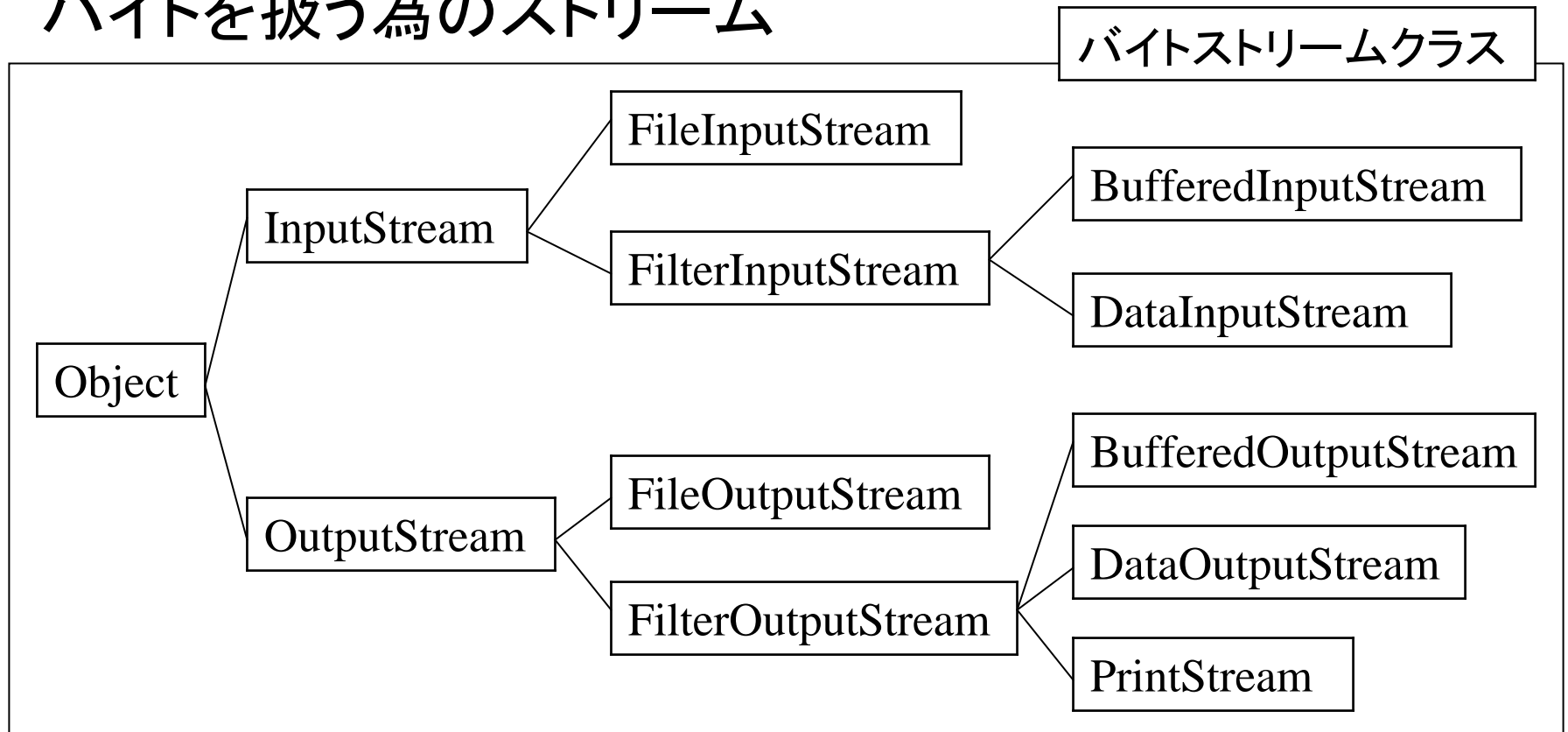
```
                pw.close();
            }
        } catch(Exception e) {
            System.out.println( "Exception: " + e );
        }
    }
}
```

## 実行結果

```
true
A
4000
45.67
45.67
Hello
99
```

# 11.5 バイトストリーム

- バイトストリームとは  
バイトを扱う為のストリーム



# OutputStream

---

## □ OutputStream クラス

すべてのバイト出力ストリームに利用できる機能が定義されている

OutputStream クラスに定義されているインスタンスメソッド

<code>void close() throw IOException</code>	出力ストリームをクローズする
<code>void write(int <i>i</i>) throw IOException</code>	ストリームに <i>i</i> の下位8ビットを書き込む

(p.339 表11-4参照)

# InputStream

---

## □ InputStream クラス

すべてのバイト入力ストリームに利用できる機能が定義されている

InputStream クラスに定義されているインスタンスメソッド

<code>void close() throw IOException</code>	入力ストリームをクローズする
<code>void read(int <i>i</i>) throw IOException</code>	入力ストリームから1バイトを読み取る

(p.342 表11-6参照)

# FileOutputStream

---

- FileOutputStream クラス
  - OutputStream を拡張したクラス
  - このクラスを使用してファイルにバイナリデータを書き込む事ができる

## FileOutputStream コンストラクタ

FileOutputStream(String ファイルパス) throw IOException

FileOutputStream(String ファイルパス, boolean 変数) throw IOException

FileOutputStream(File ファイルオブジェクト) throw IOException

# FileInputStream

---

- FileInputStream クラス
  - InputStream を拡張したクラス
  - このクラスを使用してファイルからバイナリデータを読み取る事ができる

## FileInputStream コンストラクタ

FileInputStream(String ファイルパス) throw FileNotFoundException

FileInputStream(File ファイルオブジェクト) throw FileNotFoundException

# FileOutputStream、FileInputStreamの例

```
import java.io.*;

class FOSDemo {
    public static void main(String args[]) {
        try {
            FileOutputStream fos = new
                FileOutputStream (args[0]);
            for (int i = 0 ; i < 12 ; i++) {
                fos.write(i);
            }
            fos.close();
        } catch(Exception e) {
            System.out.println("Exception: " + e );
        }
    }
}
```

```
>javac FOSDemo.java
>java FOSDemo test.txt
```

```
import java.io.*;

class FISDemo {
    public static void main(String args[]) {
        try {
            FileInputStream fis = new
                FileInputStream (args[0]);
            int i;
            while ( (i = fis.read() ) != -1 ) {
                System.out.print(i + " ");
            }
            fis.close();
        } catch(Exception e) {
            System.out.println("Exception: " + e);
        }
    }
}
```

```
>javac FISDemo.java
>java FISDemo test.txt
0 1 2 3 4 5 6 7 8 9 10 11
```

# FilterOutputStream

---

- FilterOutputStream クラス
  - OutputStream を拡張したクラス
  - このクラスは出力にフィルタをかける場合に使用する

FilterOutputStream コンストラクタ

FilterOutputStream(OutputStream 出カストリーム)

# FilterInputStream

---

- FilterInputStream クラス
  - InputStream を拡張したクラス
  - このクラスは入カストリームをフィルタにかける場合使用する

FilterInputStream コンストラクタ

FilterInputStream(InputStream 入カストリーム)

# BufferedOutputStream

---

- BufferedOutputStream クラス
  - FilterOutputStream を拡張したクラス
  - このクラスを使用してバイトストリームへの出力をバッファに入れる事ができる

BufferedOutputStream コンストラクタ

BufferedOutputStream(OutputStream 出カストリーム)

BufferedOutputStream(OutputStream 出カストリーム, int バッファサイズ)

# BufferedInputStream

---

- BufferedOutputStream クラス
- FilterInputStream を拡張したクラス
- このクラスを使用してバイトストリームからの入力をバッファに入れる事ができる

## BufferedInputStream コンストラクタ

BufferedInputStream(InputStream 入カストリーム)

BufferedInputStream(InputStream 入カストリーム, int バッファサイズ)

# BufferedOutputStream、 BufferedInputStreamの例

```
import java.io.*;

class BOSDemo {
    public static void main(String args[]) {
        try {
            FileOutputStream fos = new
                FileOutputStream (args[0]);
            BufferedOutputStream bos = new
                BufferedOutputStream (fos);
            for (int i = 0 ; i < 12 ; i++) {
                bos.write(i);
            }
            bos.close();
        } catch(Exception e) {
            System.out.println("Exception: " + e );
        }
    }
}
```

```
>javac BOSDemo.java
>java BOSDemo test2.txt
```

```
import java.io.*;

class BISDemo {
    public static void main(String args[]) {
        try {
            FileInputStream fis = new
                FileInputStream (args[0]);
            BufferedInputStream bis = new
                BufferedInputStream (fis);
            int i;
            while ( (i = bis.read() ) != -1 ) {
                System.out.print(i + " ");
            }
            fis.close();
        } catch(Exception e) {
            System.out.println("Exception: " + e);
        }
    }
}
```

```
>javac BISDemo.java
>java BISDemo test2.txt
0 1 2 3 4 5 6 7 8 9 10 11
```

# DataOutputStream

---

- ByteArrayOutputStream クラス
  - FilterOutputStream を拡張したクラス
  - DataOutput を実装
  - 基本データ型をバイト出力ストリームに書き込むことができる

OutputStream コンストラクタ

OutputStream(OutputStream 出力ストリーム)

# DataInputStream

---

- DataInputStream クラス
  - FilterInputStream を拡張したクラス
  - DataInput を実装
  - バイト入カストリームから単純なJava型を読み取る事ができる

DataInputStream コンストラクタ

DataInputStream(InputStream 入カストリーム)

## DataOutput インターフェイス、DataInput インターフェイスに定義されているインスタンスメソッド

### DataOutput インターフェイスに定義されているインスタンスメソッド

<code>void write(int <i>i</i>) throws IOException</code>	ストリームに <i>i</i> を書き込む
<code>void writeInt(int <i>i</i>) throws IOException</code>	ストリームに <i>i</i> を書き込む

(p.340-341 表11-5参照)

### DataInput インターフェイスに定義されているインスタンスメソッド

<code>void read(byte <i>buffer</i>[]) throws IOException</code>	<i>Buffer.length</i> バイトまでバッファを読み取り、バイト数を返す
<code>Void readInt() throws IOException</code>	ストリームからInt型を読み取って返す

(p.343-344 表11-7参照)

# DataOutputStream、DataInputStreamの例(1/2)

---

```
import java.io.*;
```

```
class DOSDemo {  
    public static void main(String args[]) {  
        try {  
            FileOutputStream fos = new FileOutputStream(args[0]);  
            DataOutputStream dos = new DataOutputStream(fos);  
            dos.writeBoolean(false);  
            dos.writeByte(Byte.MAX_VALUE);  
            dos.writeChar('A');  
            dos.writeInt(Integer.MAX_VALUE);  
            dos.writeLong(Long.MAX_VALUE);  
            dos.writeShort(Short.MAX_VALUE);  
            fos.close();  
        } catch(Exception e) {  
            System.out.println("Exception e: " + e);  
        }  
    }  
}
```

```
>javac DOSDemo.java
```

```
>java DOSDemo test3.txt
```

# DataOutputStream、DataInputStreamの例(2/2)

```
import java.io.*;

class DISDemo {
    public static void main(String args[]) {
        try {
            FileInputStream fis = new FileInputStream(args[0]);
            DataInputStream dis = new DataInputStream(fis);
            System.out.println(dis.readBoolean());
            System.out.println(dis.readByte());
            System.out.println(dis.readChar());
            System.out.println(dis.readInt());
            System.out.println(dis.readLong());
            System.out.println(dis.readShort());
            fis.close();
        } catch (Exception e) {
            System.out.println("Exception e: " + e);
        }
    }
}
```

```
>javac DISDemo.java
>java DISDemo test3.txt
False
127
A
2147483647
9223372936854775807
32767
```

# PrintStream

---

- PrintStream クラス
  - FilterOutputStream を拡張したクラス
  - System.out をとおして使用してきたすべての書式設定機能を提供
  - 静的変数 System.out は PrintStream

## PrintStream コンストラクタ

PrintStream(OutputStream 出力ストリーム)

PrintStream(OutputStream 出力ストリーム, boolean 変数)

PrintStream(OutputStream 出力ストリーム, String 変数)

# 宿題

---

0~255までのバイナリデータを1つのファイルに書き込むプログラムと、ファイルからバイナリデータを読み取り、2つのファイルに分割して書き込むプログラムを作成せよ。