

Entrées/sorties

Pratique de la programmation orientée-objet
Michel Schinz – 2017-02-27

Flots en Java (`java.io`)

Le flot (*stream*) est l'abstraction centrale des entrées/sorties de `java.io`. Il y en a quatre types :

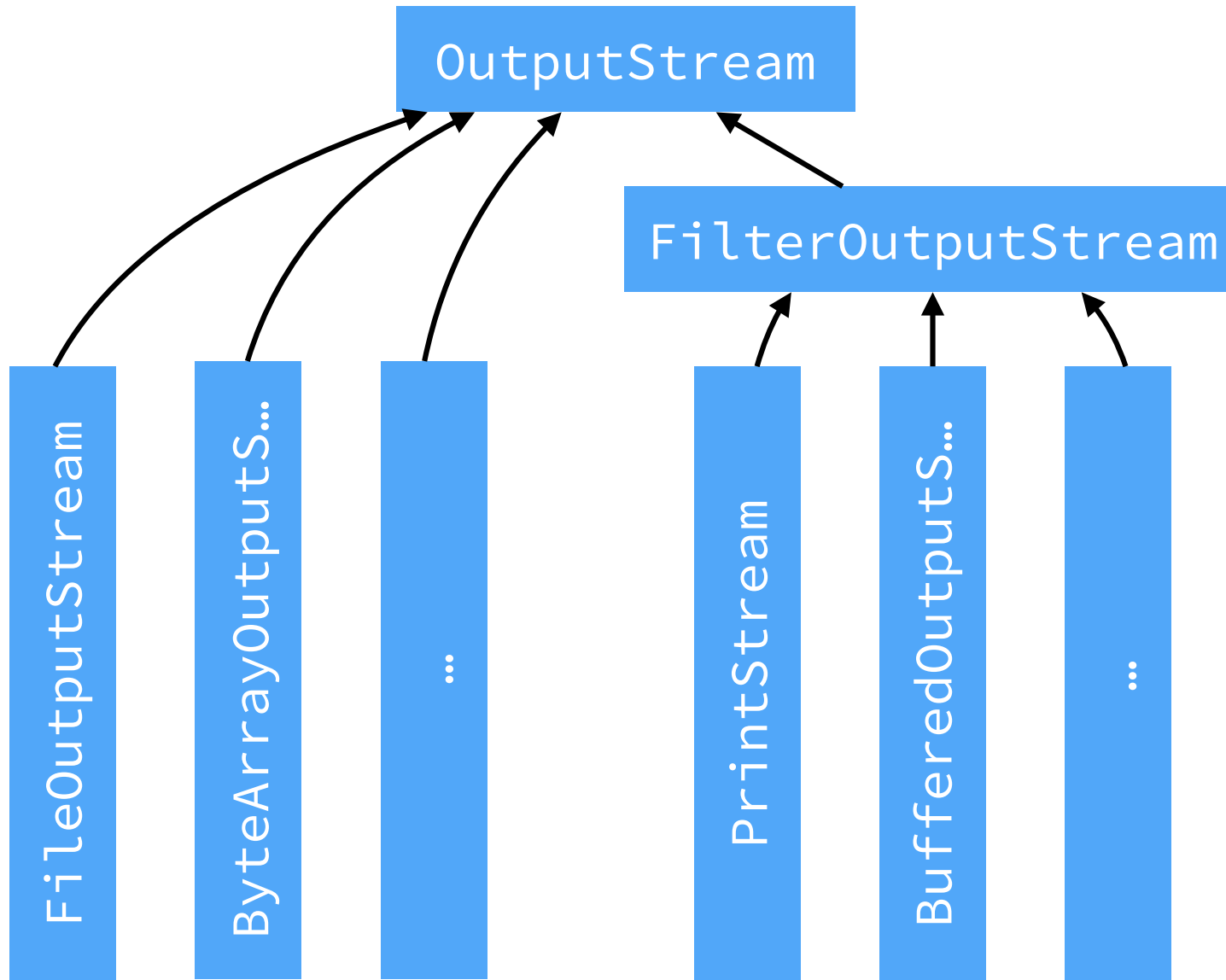
	Entrée	Sortie
Octets	<code>InputStream</code>	<code>OutputStream</code>
Caractères	<code>Reader</code>	<code>Writer</code>

A chaque type de flot, représenté par une classe abstraite, correspond une hiérarchie.

OutputStream

```
abstract public class OutputStream {  
    // Ecrit les 8 bits de poids faible de b  
    // dans le flot.  
    abstract void write(int b);  
  
    // Ecrit la totalité des octets de b dans  
    // le flot.  
    void write(byte[] b);  
  
    // Ferme le flot.  
    void close();  
}
```

Flots de sortie d'octets



InputStream

```
abstract public class InputStream {  
    // Lit et retourne le prochain octet du  
    // flot (entier entre 0 et 255) ou -1.  
    abstract int read();  
  
    // Lit et place dans le tableau b les  
    // prochains octets du flot, retourne le  
    // nombre lu ou -1.  
    int read(byte[] b);  
  
    // Ferme le flot.  
    void close();  
}
```

Représentation des caractères

ASCII (1975)

1 caractère = 7 bits (127 caractères)

PDF : fr en v · d · m	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
002	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



caractères de contrôle



caractères imprimables

ISO 8859-1 (1985)

1 caractère = 8 bits (256 caractères)

ISO-8859-1																
	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1-	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2-	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	007F
8-	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTIS	FLD	FLU	RI	SSZ	SSJ
	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	008C	008D	008E	008F
9-	DCS	FUL	FU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	SW	APC
	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
A-	NSBP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	—
	00A0	00A1	00A2	00A3	00A4	00A5	00A6	00A7	00A8	00A9	00AA	00AB	00AC	00AD	00AE	00AF
B-	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
	00B0	00B1	00B2	00B3	00B4	00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF
C-	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
	00C0	00C1	00C2	00C3	00C4	00C5	00C6	00C7	00C8	00C9	00CA	00CB	00CC	00CD	00CE	00CF
D-	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
	00D0	00D1	00D2	00D3	00D4	00D5	00D6	00D7	00D8	00D9	00DA	00DB	00DC	00DD	00DE	00DF
E-	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
	00E0	00E1	00E2	00E3	00E4	00E5	00E6	00E7	00E8	00E9	00EA	00EB	00EC	00ED	00EE	00EF
F-	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
	00F0	00F1	00F2	00F3	00F4	00F5	00F6	00F7	00F8	00F9	00FA	00FB	00FC	00FD	00FE	00FF
	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F

ASCII

Unicode (1987-...)

1 caractère = 20 bits (~1 million de caractères)

	000	001	002	003	004	005	006	007
0	NUL	SOH	SP	0	@	P	`	p
1	DC1	DC1	!	1	A	Q	a	q
2	DC2	DC2	"	2	B	R	b	r
3	DC3	DC3	#	3	C	S	c	s
4	DC4	DC4	\$	4	D	T	d	t
5	DC5	DC5	%	5	E	U	e	u
6	DC6	DC6	&	6	F	V	f	v
7	DC7	DC7	'	7	G	W	g	w
8	DC8	DC8	(8	H	X	h	x
9	DC9	DC9)	9	I	Y	i	y
A	DC10	DC10	*	:	J	Z	j	z
B	DC11	DC11	+	;	K	[k	{
C	DC12	DC12	,	<	L	\	l	
D	DC13	DC13	-	=	M]	m	}
E	DC14	DC14	.	>	N	^	n	~
F	DC15	DC15	/	?	O	_	o	DEL

ASCII

	008	009	00A	00B	00C	00D	00E	00F
0	DC16	DC16	DC16	°	À	Ð	à	ð
1	DC17	DC17	±	Á	Ñ	á	ñ	
2	DC18	DC18	²	Â	Ò	â	ò	
3	DC19	DC19	£	Ã	Ó	ã	ó	
4	DC20	DC20	¤	Ä	Ô	ä	ô	
5	DC21	DC21	¥	Å	Ö	å	ö	
6	DC22	DC22	¦	Æ	Ø	æ	ø	
7	DC23	DC23	§	·	Ç	×	ç	÷
8	DC24	DC24	¨	È	Ø	è	ø	
9	DC25	DC25	©	É	Ù	é	ù	
A	DC26	DC26	ª	Ê	Ú	ê	ú	
B	DC27	DC27	«	Ë	Û	ë	û	
C	DC28	DC28	¼	Ì	Ü	ì	ü	
D	DC29	DC29	½	Í	Ý	í	ý	
E	DC30	DC30	¾	Î	Þ	î	þ	
F	DC31	DC31	¿	Ï	ß	ï	ÿ	

...










ISO 8859-1

Unicode (1987-...)

2A00

Supplemental Mathematical Operators

2AFF

	2A0	2A1	2A2	2A3	2A4	2A5	2A6	2A7	2A8	2A9	2AA	2AB	2AC	2AD	2AE	2AF
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

Unicode (1987-...)

13000

Egyptian Hieroglyphs

130DF

	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	130A	130B	130C	130D
0														
1														
2														
3														
4														
5														
6														
7														
8														
9														
A														
B														
C														
D														
E														
F														

Unicode (1987-...)

1F600

Emoticons

1F64F

	1F600	1F601	1F602	1F603	1F604
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
A					
B					
C					
D					
E					
F					

Reader

```
abstract public class Reader {  
    // Lit et retourne le prochain caractère du  
    // flot (entier entre 0 et  $2^{16}-1$ ) ou -1.  
    int read();  
    // Lit et place dans le tableau c les  
    // prochains caractères du flot, retourne  
    // le nombre lu ou -1.  
    int read(char[] c);  
    // Ferme le flot.  
    void close();  
}
```

Writer

```
abstract public class Writer {  
    // Ecrit les 16 bits de poids faible de c  
    // dans le flot.  
    void write(int c);  
    // Ecrit la totalité des caractères de a  
    // dans le flot.  
    void write(char[] a);  
    // Ferme le flot.  
    void close();  
}
```