

Entrées/sorties

Pratique de la programmation orientée-objet
Michel Schinz

Flots en Java (java.io)

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

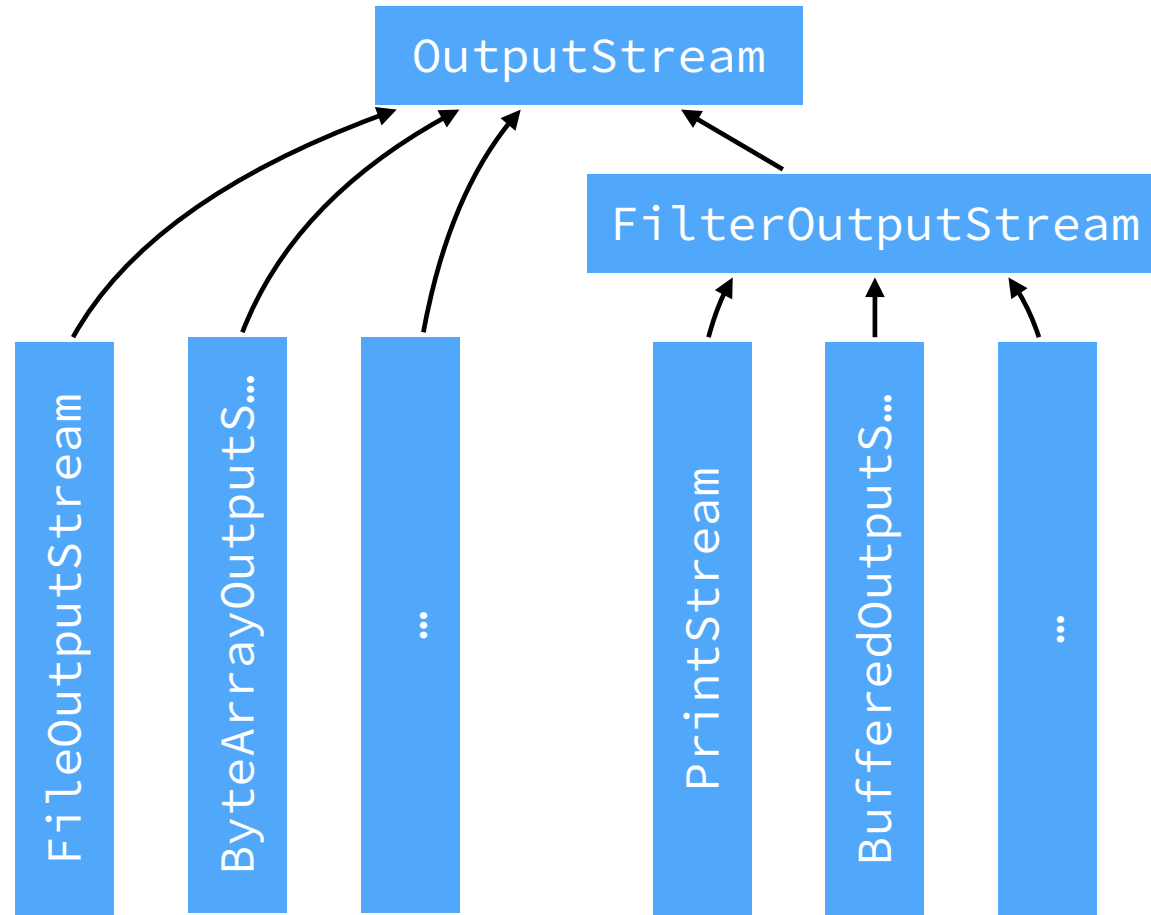
	Entrée	Sortie
Octets	InputStream	OutputStream
Caractères	Reader	Writer

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, à partir de la  
    // position o, les l prochains octets du flot.  
    // Retourne le nombre d'octets lus.  
    int readNBytes(byte[] b, int o, int l);  
  
    // Ferme le flot.  
    void close();  
}
```

Représentation des caractères

ASCII (1975)

1 caractère = 7 bits (128 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-	0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1-	0010	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
2-	0020	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
3-	0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
4-	0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	
5-	0050	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	
6-	0060	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	
7-	0070	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	
8-	0080	SPAD	MOB	BMH	NHM	IND	NEL	SA	SA	HTS	HTJ	VTS	PLD	PLJ	RZ	RSZ	SSZ
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	
9-	0090	DCS	FUL	FU2	STB	CH	MR	SA	SA	RS	SC1	SC2	CS1	ST	OPC	PM	ASC
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	
A-	00A0	MSB	!	¢	£	¤	¥	¦	§	¨	©	*	«	¬	®	¯	—
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	
B-	00B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	
C-	00C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	
D-	00D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	
E-	00E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	
F-	00F0	š	š	š	š	š	š	š	š	š	š	š	š	š	š	š	š
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	

↑
ASCII
↓

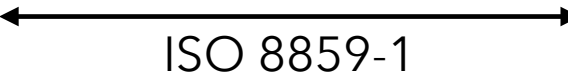
Unicode (1987-...)

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

	000	001	002	003	004	005	006	007
0	[NUL]	[SOH]	[STX]	[ETX]	[EOT]	[BELL]	[BS]	[TAB]
1	[LF]	[VT]	[FF]	[CR]	[SH]	[ESC]	[F1]	[F2]
2	[F3]	[F4]	[F5]	[F6]	[F7]	[F8]	[F9]	[F10]
3	[F11]	[F12]	[F13]	[F14]	[F15]	[F16]	[F17]	[F18]
4	[F19]	[F20]	[F21]	[F22]	[F23]	[F24]	[F25]	[F26]
5	[F27]	[F28]	[F29]	[F30]	[F31]	[F32]	[F33]	[F34]
6	[F35]	[F36]	[F37]	[F38]	[F39]	[F40]	[F41]	[F42]
7	[F43]	[F44]	[F45]	[F46]	[F47]	[F48]	[F49]	[F50]
8	[F51]	[F52]	[F53]	[F54]	[F55]	[F56]	[F57]	[F58]
9	[F59]	[F60]	[F61]	[F62]	[F63]	[F64]	[F65]	[F66]
A	[F67]	[F68]	[F69]	[F70]	[F71]	[F72]	[F73]	[F74]
B	[F75]	[F76]	[F77]	[F78]	[F79]	[F80]	[F81]	[F82]
C	[F83]	[F84]	[F85]	[F86]	[F87]	[F88]	[F89]	[F90]
D	[F91]	[F92]	[F93]	[F94]	[F95]	[F96]	[F97]	[F98]
E	[F99]	[F100]	[F101]	[F102]	[F103]	[F104]	[F105]	[F106]
F	[F107]	[F108]	[F109]	[F110]	[F111]	[F112]	[F113]	[F114]

	008	009	00A	00B	00C	00D	00E	00F
0	[G0]	[G1]	[G2]	[G3]	[G4]	[G5]	[G6]	[G7]
1	[G8]	[G9]	[GA]	[GB]	[GC]	[GD]	[GE]	[GF]
2	[GG]	[GH]	[GI]	[GJ]	[GK]	[GL]	[GM]	[GN]
3	[GO]	[GP]	[GQ]	[GR]	[GS]	[GT]	[GU]	[GV]
4	[GW]	[GX]	[GY]	[GZ]	[GA]	[GB]	[GC]	[GD]
5	[GE]	[GF]	[GG]	[GH]	[GI]	[GJ]	[GK]	[GL]
6	[GM]	[GN]	[GO]	[GP]	[GQ]	[GR]	[GS]	[GT]
7	[GU]	[GV]	[GW]	[GX]	[GY]	[GZ]	[GA]	[GB]
8	[GC]	[GD]	[GE]	[GF]	[GG]	[GH]	[GI]	[GJ]
9	[GK]	[GL]	[GM]	[GN]	[GO]	[GP]	[GQ]	[GR]
A	[GS]	[GT]	[GU]	[GV]	[GW]	[GX]	[GY]	[GZ]
B	[GA]	[GB]	[GC]	[GD]	[GE]	[GF]	[GG]	[GH]
C	[GI]	[GJ]	[GK]	[GL]	[GM]	[GN]	[GO]	[GP]
D	[GR]	[GS]	[GT]	[GU]	[GV]	[GW]	[GX]	[GY]
E	[GZ]	[GA]	[GB]	[GC]	[GD]	[GE]	[GF]	[GG]
F	[GH]	[GI]	[GJ]	[GK]	[GL]	[GM]	[GN]	[GO]

...



Unicode (1987-...)

	2A0	2A1	2A2	2A3	2A4	2A5	2A6	2A7	2A8	2A9	2AA	2AB	2AC	2AD	2AE	2AF
0	2A00	2A01	2A02	2A03	2A04	2A05	2A06	2A07	2A08	2A09	2AA0	2AB0	2AC0	2AD0	2AE0	2AF0
1	2A01	2A11	2A21	2A31	2A41	2A51	2A61	2A71	2A81	2A91	2AA1	2AB1	2AC1	2AD1	2AE1	2AF1
2	2A02	2A12	2A22	2A32	2A42	2A52	2A62	2A72	2A82	2A92	2AA2	2AB2	2AC2	2AD2	2AE2	2AF2
3	2A03	2A13	2A23	2A33	2A43	2A53	2A63	2A73	2A83	2A93	2AA3	2AB3	2AC3	2AD3	2AE3	2AF3
4	2A04	2A14	2A24	2A34	2A44	2A54	2A64	2A74	2A84	2A94	2AA4	2AB4	2AC4	2AD4	2AE4	2AF4
5	2A05	2A15	2A25	2A35	2A45	2A55	2A65	2A75	2A85	2A95	2AA5	2AB5	2AC5	2AD5	2AE5	2AF5
6	2A06	2A16	2A26	2A36	2A46	2A56	2A66	2A76	2A86	2A96	2AA6	2AB6	2AC6	2AD6	2AE6	2AF6
7	2A07	2A17	2A27	2A37	2A47	2A57	2A67	2A77	2A87	2A97	2AA7	2AB7	2AC7	2AD7	2AE7	2AF7
8	2A08	2A18	2A28	2A38	2A48	2A58	2A68	2A78	2A88	2A98	2AA8	2AB8	2AC8	2AD8	2AE8	2AF8
9	2A09	2A19	2A29	2A39	2A49	2A59	2A69	2A79	2A89	2A99	2AA9	2AB9	2AC9	2AD9	2AE9	2AF9
A	2A0A	2A1A	2A2A	2A3A	2A4A	2A5A	2A6A	2A7A	2A8A	2A9A	2AA0	2ABA	2ACA	2ADA	2AEA	2AFA
B	2A0B	2A1B	2A2B	2A3B	2A4B	2A5B	2A6B	2A7B	2A8B	2A9B	2AAB	2ABB	2ACB	2ADB	2AEB	2AFB
C	2A0C	2A1C	2A2C	2A3C	2A4C	2A5C	2A6C	2A7C	2A8C	2A9C	2AAC	2ABC	2ACC	2ADC	2AEC	2AFC
D	2A0D	2A1D	2A2D	2A3D	2A4D	2A5D	2A6D	2A7D	2A8D	2A9D	2AAD	2ABD	2ACD	2ADD	2AED	2AFD
E	2A0E	2A1E	2A2E	2A3E	2A4E	2A5E	2A6E	2A7E	2A8E	2A9E	2AAE	2ABE	2ACE	2ADE	2AEE	2AFE
F	2A0F	2A1F	2A2F	2A3F	2A4F	2A5F	2A6F	2A7F	2A8F	2A9F	2AAF	2ABF	2ACF	2ADF	2AEF	2AFF

Unicode (1987-...)

	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	130A	130B	130C	130D
0	13000	13010	13020	13030	13040	13050	13060	13070	13080	13090	130A0	130B0	130C0	130D0
1	13001	13011	13021	13031	13041	13051	13061	13071	13081	13091	130A1	130B1	130C1	130D1
2	13002	13012	13022	13032	13042	13052	13062	13072	13082	13092	130A2	130B2	130C2	130D2
3	13003	13013	13023	13033	13043	13053	13063	13073	13083	13093	130A3	130B3	130C3	130D3
4	13004	13014	13024	13034	13044	13054	13064	13074	13084	13094	130A4	130B4	130C4	130D4
5	13005	13015	13025	13035	13045	13055	13065	13075	13085	13095	130A5	130B5	130C5	130D5
6	13006	13016	13026	13036	13046	13056	13066	13076	13086	13096	130A6	130B6	130C6	130D6
7	13007	13017	13027	13037	13047	13057	13067	13077	13087	13097	130A7	130B7	130C7	130D7
8	13008	13018	13028	13038	13048	13058	13068	13078	13088	13098	130A8	130B8	130C8	130D8
9	13009	13019	13029	13039	13049	13059	13069	13079	13089	13099	130A9	130B9	130C9	130D9
A	1300A	1301A	1302A	1303A	1304A	1305A	1306A	1307A	1308A	1309A	130AA	130BA	130CA	130DA
B	1300B	1301B	1302B	1303B	1304B	1305B	1306B	1307B	1308B	1309B	130AB	130BB	130CB	130DB
C	1300C	1301C	1302C	1303C	1304C	1305C	1306C	1307C	1308C	1309C	130AC	130BC	130CC	130DC
D	1300D	1301D	1302D	1303D	1304D	1305D	1306D	1307D	1308D	1309D	130AD	130BD	130CD	130DD
E	1300E	1301E	1302E	1303E	1304E	1305E	1306E	1307E	1308E	1309E	130AE	130BE	130CE	130DE
F	1300F	1301F	1302F	1303F	1304F	1305F	1306F	1307F	1308F	1309F	130AF	130BF	130CF	130DF

Unicode (1987-...)

	1F60	1F61	1F62	1F63	1F64
0	1F600	1F610	1F620	1F630	1F640
1	1F601	1F611	1F621	1F631	1F641
2	1F602	1F612	1F622	1F632	1F642
3	1F603	1F613	1F623	1F633	1F643
4	1F604	1F614	1F624	1F634	1F644
5	1F605	1F615	1F625	1F635	1F645
6	1F606	1F616	1F626	1F636	1F646
7	1F607	1F617	1F627	1F637	1F647
8	1F608	1F618	1F628	1F638	1F648
9	1F609	1F619	1F629	1F639	1F649
A	1F60A	1F61A	1F62A	1F63A	1F64A
B	1F60B	1F61B	1F62B	1F63B	1F64B
C	1F60C	1F61C	1F62C	1F63C	1F64C
D	1F60D	1F61D	1F62D	1F63D	1F64D
E	1F60E	1F61E	1F62E	1F63E	1F64E
F	1F60F	1F61F	1F62F	1F63F	1F64F

UTF-8

Encodage d'Unicode à taille variable (1 à 4 octets par caractère).

Plage	octet 1	octet 2	octet 3	octet 4
0 à 7F	0xxxxxxx			
... à 7FF	110xxxxx	10xxxxxx		
... à FFFF	1110xxxx	10xxxxxx	10xxxxxx	
... à 10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-16

Encodage d'Unicode à taille variable (2 ou 4 octets par caractère) :

- de 0 à D7FF et de E000 à FFFF : représentés tels quels,
- de 10000 à 10FFFF : représentés par 4 octets, au moyen de *surrogate pairs*.

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();  
}
```