

EXERCICE 06

Thème

Développement logiciel: Les caractères utilisables dans un identificateur Java

Pour vous rappeler ce qu'est un identificateur, faites une recherche « identifier vs literal »

Cahier des charges

Afficher à l'écran une table de tous les caractères utilisables dans un identificateur Java. Pour chaque caractère on affichera son symbole alphanumérique s'il existe, et sa valeur ASCII en décimal et en hexadécimal

'0'	30	48	'1'	31	49	'2'	32	50	'3'	33	51
'4'	34	52	'5'	35	53	'6'	36	54	'7'	37	55
'8'	38	56	'9'	39	57	'A'	41	65	'B'	42	66
'C'	43	67	'D'	44	68	'E'	45	69	'F'	46	70
'G'	47	71	'H'	48	72	'I'	49	73	'J'	4a	74
'K'	4b	75	'L'	4c	76	'M'	4d	77	'N'	4e	78
'O'	4f	79	'P'	50	80	'Q'	51	81	'R'	52	82
'S'	53	83	'T'	54	84	'U'	55	85	'V'	56	86
'W'	57	87	'X'	58	88	'Y'	59	89	'Z'	5a	90
'_'	5f	95	'a'	61	97	'b'	62	98	'c'	63	99
'd'	64	100	'e'	65	101	'f'	66	102	'g'	67	103

Renseignements techniques:

A chaque type de donnée primitive, *int*, *char*, ... correspond une classe (Wrapper) *Integer*, *Character*, .. Ces classes offrent des méthodes statiques spécifiques à chacune de ces données. Ainsi par exemple:

`Character.toUpperCase('x')` retourne *true* si *x* est une majuscule
`Integer.toHexString(x)` retourne une chaîne de caractères correspondant à la valeur hexadécimale de *x*

`Character.isJavaIdentifierPart('x')` indique si le caractère peut faire partie d'un identifieur

Dans la Javadoc, voir aussi `Character.isISOControls`

Sur google, faites une recherche « identifier vs literal »

Documents

Ce développement fera l'objet d'un dossier de réalisation logicielle comprenant

- ☞ Une spécification textuelle (pseudo-code)
- ☞ Un document de conception détaillée sous forme d'un PDL ou d'un organigramme
- ☞ Un source correctement documenté