

Ogólny format specyfikacji przekształceń używany funkcji printf

%[znacznik][minimalna-długość-pola][.][precyzja]specyfikator-formatu

Specyfikatory formatowania funkcji printf dla języka Java i C

Specyfikatory			Opis
Java	C i C++		
d	h, l	d, i	liczba całkowita ze znakiem w postaci dziesiętnej
	h, l	u	liczba całkowita bez znaku w postaci dziesiętnej
o		o	liczba całkowita bez znaku w postaci ósemkowej
x	h, l	x	liczba całkowita bez znaku w postaci szesnastkowej (małe litery)
	h, l	X	liczba całkowita bez znaku w postaci szesnastkowej (wielkie litery)
f	l, L	f	liczba zmiennoprzecinkowa z separatorem dziesiętnym
e	l, L	e	liczba zmiennoprzecinkowa w zapisie wykładniczym (mała litera e)
	l, L	E	liczba zmiennoprzecinkowa w zapisie wykładniczym (duża litera E)
a		a	liczba zmiennoprzecinkowa w zapisie wykładniczym szesnastkowym (mała litera p)
		A	liczba zmiennoprzecinkowa w zapisie wykładniczym szesnastkowym (duża litera P)
g	l, L	g	krótsza z notacji %e lub %f
	l, L	G	krótsza z notacji %E lub %f
c		c	znak
s		s	ciąg znaków (łańcuch)
		P, p	odpowiadający wskaźnikowi adres w postaci szesnastkowej
n			separator wiersza właściwy dla platformy
		N, n	wskaźnik do liczby całkowitej, w której zostanie zapamiętana liczba dotychczas wypisanych znaków
%		%	wypisanie znaku %
b			wartość logiczna
h			kod mieszający
tx			data i godzina

Przedrostki specyfikatorów dla języka C

Przedrostki			Opis
Java	C	C++	
	l		liczba całkowita long
	h		liczba całkowita short
	l		liczba zmiennoprzecinkowa double
	L		liczba zmiennoprzecinkowa long double

Rodzaje znaczników dla języka Java i C

Znaczniki			Opis
Java	C i C++		
+	+	liczby dodatnie poprzedzone są znakiem plus	
spacja	spacja	liczby dodatnie poprzedzone są spacją	
0	0	pole wypełnione jest początkowymi zerami	
-	-	wartość wyjściowa jest wyrównywana w obrębie pola do lewej strony	
(liczby ujemne prezentowane są w nawiasach	
,		poszczególne grupy są rozdzielone (po trzy znaki)	
#	#	liczby zmiennopozycyjne formatowane są z użyciem kropki dziesiętnej	
##	##	liczby zmiennopozycyjne szesnastkowe formatowane są z prefiksem 0X lub 0x, natomiast liczby zmiennopozycyjne ósemkowe formatowane są z prefiksem 0	
\$		indeks argumentu sformatowania	
<		formatowanie dla argumentu funkcji printf jak już poprzednio (wcześniej) sformatowany	

Znaczenie pola „minimalna długość pola” dla języka Java i C

Zastosowanie dla:	Opis
Wszystkie oprócz: %, p, c	Jeśli wypisywany element jest krótszy niż określona minimalna długość pola, podczas wypisywania element zostaje uzupełniony do odpowiedniej długości spacjami. Jeśli jednak wynik wykracza poza określony rozmiar nie jest on skracany.

Znaczenie pola „precyzja” dla języka Java i C

Zastosowanie dla:	Opis
d, i, o, x, u	Określa ile cyfr powinno zostać wyświetlonych. W razie potrzeby są dodawane zera początkowe.
f, e, E, a, A	Określa liczbę cyfr wyświetlanych za separatorem dziesiętnym lub szesnastkowym.
g, G	Określa liczbę cyfr znaczących.
s	Określa maksymalną długość pola. Jeśli napis będzie wykraczał poza określoną długość zostanie skrócony.

Sposób wywołania funkcji printf dla języka Java i C

Java	C i C++
System.out.printf("Liczba całkowita: %d",a);	printf("Liczba całkowita: %d",a);

Przykłady użycia funkcji printf

Instrukcja	Ekran
printf("Napis: Ala ma kota.");	Napis: Ala ma kota.
int a = 5; printf("%d", a);	5
int a = 12; printf("Liczba dziesiętna: %d", a);	Liczba dziesiętna: 12
int a = 12; printf("Liczba dziesiętna: %06d", a);	Liczba dziesiętna: 000012
int a = 12; printf("Liczba osemkowa: %o", a);	Liczba osemkowa: 14
int a = 12; printf("Liczba szesnastkowa: %x", a);	Liczba szesnastkowa: c
double b = 5.67; printf("Liczba zmiennoprzecinkowa: %f", b);	Liczba zmiennoprzecinkowa: 5.670000
double b = 5.67; printf("Liczba zmiennoprzecinkowa: %08.3f", b);	Liczba zmiennoprzecinkowa: 0005.670
double b = 47.67; printf("Liczba w zapisie wykładniczym: %e", b);	Liczba w zapisie wykładniczym: 4.767000e+01
double b = 47.67; printf("Zapis wykładniczy szesnastkowy: %a", b);	Zapis wykładniczy szesnastkowy: 0x1.7d5c28f5c28f6p5
double b = 47.67; printf("Zapis wykładniczy szesnastkowy: %.3a", b);	Zapis wykładniczy szesnastkowy: 0x1.7d6p5
int a = 12; double b = 5.67; int c = 302; printf("Liczba: %d, oraz %f, oraz %d.", a, b, c);	Liczba: 12, oraz 5,670000, oraz 302.
char z = 'A'; printf("Znak: %c",z);	Znak: A
char l[] = „Ala ma kota.”; //Dla języka C printf("Napis: %s", l);	Napis: Ala ma kota.

Stale znakowe

Stale znakowe		Opis	Stale znakowe		Opis
Ozn.	Zapis		Ozn.	Zapis	
NL(LF)	\n	nowy wiersz	\	\\	kreska ukośna w lewo
HT	\t	tabulacja pozioma	?	\?	znak zapytania
VT	\v	tabulacja pionowa	‘	\‘	apostrof
BS	\b	cofanie	”	\”	cudzysłów
CR	\r	powrót karetki	ooo	\ooo	liczba ósemkowa
FF	\f	nowa strona	hh	\xhh	liczba szesnastkowa
BEL	\a	alarm	NUL	\0	wartość null