



ED Estrutura de Dados e Armazenamento

Saída Formatada

Profa. Célia Taniwaki

String.format ou System.out.printf

 O método format da classe String é utilizado para formatar uma String.

 O método printf foi implementado no Java para manter a compatibilidade com as linguagens C e C++

 Ambos os métodos recebem como argumento uma String que é a mensagem formatada. No lugar onde aparecerão os dados, dentro dessa String, colocamos os formatos que deverão ser aplicados aos dados. Após a String com a mensagem formatada, seguem as variáveis separadas por vírgulas.

Exemplo:

Formatos:

Tipo	Descrição
%c	caracter (tipo char ou Character)
% d	número inteiro (tipo int ou Integer)
	número real, com casas decimais (tipos float ou Float, double
%f	ou Double)
%s	String
%b	booleano (boolean ou Boolean)
%. 2 f	número com casas decimais, com 2 dígitos após a vírgula

^{*} Em negrito já utilizamos no semestre anterior

Formatos

É possível usar esses formatos para fazer com que o valor fique em um espaço de XX caracteres:

%2d alinhado à direita, ocupando um espaço de 2 dígitos

Código: System.out.printf("%2d", 7);

Saída: _7 (um espaço em branco e o 7)

Mas não limita em 2 espaços se tiver mais do que 3 dígitos

Código: System.out.printf("%2d", 123);

Saída: 123

%-2d alinhado à esquerda, ocupando um espaço de 2 dígitos

Código: System.out.printf("%-2d", 7);

Saída: 7 (7 e o espaço em branco)

%6d alinhado à direita, ocupando um espaço de 6 dígitos preenche com brancos à esquerda se tiver menos de 6 dígitos

Código: System.out.printf("%6d", 123);
Saída: 123 (3 espaços em branco e o 123)

%-6d alinhado à esquerda, ocupando um espaço de 6 dígitos, preenche com brancos à direita se tiver menos de 6 dígitos

Código: System.out.printf("%-6d", 123);

Saída: 123____ (123 e 3 espaços em branco)

%06d alinhado à direita, ocupando um espaço de 6 dígitos preenche com zeros à esquerda se tiver menos de 6 dígitos

Código: System.out.printf("%06d", 123);

Saída: 000123 (3 zeros e o 123)

O mesmo vale para **Strings**:

%7s alinhado à direita, ocupando um espaço de 7 caracteres preenche com brancos à esquerda se tiver menos de 7

Código: System.out.printf("%7s", "Maria");

Saída: __Maria (2 espaços em branco e "Maria")

%-7s alinhado à esquerda, ocupando um espaço de 7 caracteres preenche com brancos à direita se tiver menos de 7 caracteres

Código: System.out.printf("%-7s", "Maria");

Saída: Maria__ ("Maria" e 2 espaços em branco)

Strings (continuação):

%7S alinhado à direita, ocupando um espaço de 7 caracteres preenche com brancos à esquerda se tiver menos de 7. Como o S está maiúsculo, transforma a string para caixa alta

Código: System.out.printf("%75", "Maria");

Saída: __MARIA (2 espaços em branco e "MARIA")

%-7.3s alinhado à esquerda, ocupando um espaço de 7 caracteres, porêm exibe um máximo de 3 caracteres

Código: System.out.printf("%-7.3s", "Maria");

Saída: Mar____ ("Mar" e 2 espaços em branco)

O mesmo vale para **Double**:

%7.2f alinhado à direita, ocupando um espaço de 7 caracteres, com 2 casas decimais (os 7 caracteres incluem a vírgula).

Código: System.out.printf("%7.2f", 8.5);

Saída: ____8,50

%07.2f alinhado à direita, ocupando um espaço de 7 caracteres, com 2 casas decimais (os 7 caracteres incluem a vírgula) preenche com zeros à esquerda.

Código: System.out.printf("%07.2f", 8.5);

Saída: 0008,50

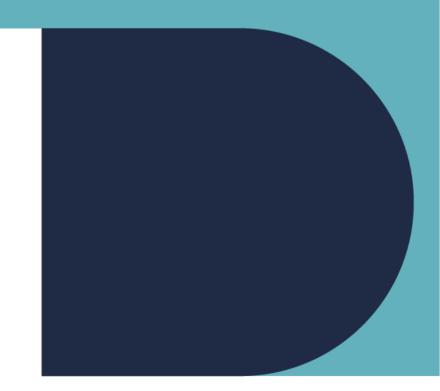
Agradeço a sua atenção!



Em caso de dúvidas, entre em contato com:

celia.taniwaki@sptech.school

giuliana.franca@sptech.school



SÃO PAULO TECH SCHOOL