



Congratulations! You passed!

Next Item



1. Uma expressão lambda pode ser utilizada no lugar de:

0 / 1 point



Um método

This should not be selected

Veja a aula Conhecendo Expressões Lambda



Uma classe qualquer



Um atributo



Uma função



Uma implementação de uma interface



2. São expressões lambda corretas (marque TODAS as corretas):

1 / 1 point



1 Consumer<String> c = (String s)->{System.out.println(s);};

Correct

Essa é a sintaxe completa que define o tipo do parâmetro e coloca os comandos entre chaves



1 Consumer<String> c = s :: System.out.println(s);

Un-selected is correct



1 Consumer<String> c = s -> System.out.println(s);

Correct

Essa sintaxe elimina a definição do tipo e as chaves por haver apenas um comando



1 Consumer<String> c = (String)->{System.out::println(s)};

Un-selected is correct



3. Uma interface para ser considerada uma interface funcional

1 / 1 point



Precisa possuir a anotação @FunctionalInterface



Pode possuir vários métodos abstratos



Precisa da anotação @Lambda no método



Precisa ter somente um método abstrato

Correct

Esse é o único requisito. Vale ressaltar que a interface pode ter outros métodos default.



4. "Só é possível utilizar expressões lambda para interfaces que já existem na API básica da linguagem Java". Essa frase é:

1 / 1 point



Verdadeira



Falsa

Correct

É possível para qualquer interface considerada funcional.



5. O que são referências de método (Method References)?

0 / 1 point



Permitem que uma interface com vários métodos seja utilizada como interface funcional

This should not be selected

Veja aula Referências de Métodos



Representação estática de um método



Substitui expressões lambda quando apenas a chamada de um método é necessária



6. Qual seria a sintaxe correta para a criação de um Method Reference para o método getNome() da classe Pessoa?

1 / 1 point



Pessoa::getNome

Correct

Essa é a sintaxe correta



Pessoa::getNome()



Pessoa->getNome()



Pessoa->getNome

