Composição de funções

Fácil

1) O que significa a composição de funções em um contexto de programação funcional?
a) Combinar duas funções em uma só
b) Criar uma função que não usa parâmetros
c) Deletar uma função
d) Chamar funções sequencialmente
RESPOSTA: A

2) Qual das seguintes linguagens de programação suporta composição de funções de forma nativa?
a) Python
b) Java
c) C
d) Assembly
RESPOSTA: A

3) Qual é a principal vantagem da composição de funções?
a) Aumentar a complexidade do código
b) Reutilizar código de forma mais eficiente
c) Reduzir a legibilidade do código
d) Aumentar o número de variáveis
RESPOSTA: B

4) A operação de composição de funções é comumente usada em:
a) Orientação a objetos
b) Programação imperativa
c) Programação funcional
d) Nenhuma das anteriores
RESPOSTA: C

5) Em Python, qual biblioteca é frequentemente usada para composição de funções?
a) NumPy
b) itertools
c) functools
d) pandas
RESPOSTA: C

6) Qual é a característica de uma função composta?
a) Ela sempre deve retornar um valor diferente
b) Ela é composta apenas de funções puras
c) Ela pode depender de variáveis globais
d) Ela pode ter efeitos colaterais
RESPOSTA: B

7) Qual das seguintes expressões representa a composição de duas funções em uma linguagem funcional?
a) f + g
b) f - g
c) f(g(x))
d) nenhuma das anteriores
RESPOSTA: C

8) No contexto da programação funcional, funções puras são importantes porque:
a) Elas não podem ser compostas
b) Elas têm efeitos colaterais
c) Elas permitem composição sem efeitos colaterais
d) Elas são mais rápidas que funções impuras
RESPOSTA: C

9) Dada a função *h(x)=2x* e *g(x)=x+3*, o que é h(g(5))?
a) 13
b) 16
c) 14
d) 10
RESPOSTA: B

10) Qual é o resultado da expressão f(g(x)) se f(x)=x+2 e g(x) = x^2?
a) x^2+2
b) 2x
c) x+2
d) 2x+2
RESPOSTA: A

Média

1) O que o método *map* pode fazer em uma composição de funções?
a) Aplicar uma função a cada elemento de um array
b) Combinar dois arrays
c) Criar um novo array vazio
d) Ordenar um array
RESPOSTA: A

2) O que a seguinte composição retorna?
```
const f = x => x + 1;
const g = x => x * 2;
const h = compose(f, g);
console.log(h(5));
```
a) 10
b) 11
c) 12
d) 6
RESPOSTA: B

3) Se você tiver uma lista de números e quiser compor uma função para dobrar e depois somar 1, qual seria a abordagem correta?
a) numbers.map(double).map(increment)
b)numbers.map(increment).map(double)
c)numbers.map(compose(increment, double))
d)compose(double, increment)(numbers)
RESPOSTA: C

4)O que a função *pipe* faz em comparação com *compose*?
a)Inverte a ordem das funções
b)Executa as funções em paralelo
c)Executa as funções da esquerda para a direita
d)Executa as funções da direita para a esquerda
RESPOSTA: A

5)O que acontece quando você compõe funções que não são puras?
O resultado será sempre o mesmo.
B) O comportamento se torna previsível.
C) O resultado pode variar dependendo do estado global.
D) A composição falha.
RESPOSTA: C

6)Qual a saída do seguinte código?
```
const toUpperCase = str => str.toUpperCase();
const exclaim = str => str + '!';
const excited = compose(toUpperCase, exclaim);
console.log(excited("hello"));
```
A) "hello!"
B) "HELLO!"
C) "HELLO!!"
D) "hello!!"
RESPOSTA: B

7)Qual a saída do código abaixo?
```
const square = x => x * x;
const cube = x => x * x * x;
const combined = x => square(cube(x));
console.log(combined(2));
```
A) 8
B) 64
C) 16
D) 32
RESPOSTA: B

8)Qual o resultado do código abaixo?
```
const add = x => x + 10;
const square = x => x * x;
const process = x => square(add(x));
```

console.log(process(5));
A) 25
B) 225
C) 75
D) 100
RESPOSTA: B

9)O que acontece quando você compõe uma função que altera o estado global?
A) A função é executada normalmente.
B) O estado global é sempre alterado.
C) A função não pode ser composta.
D) Pode causar efeitos colaterais indesejados.
RESPOSTA: D

10)Qual é o resultado da expressão
const add = x => x + 2
const multiply = x => x * 3
const composed = x => multiply(add(x))
console.log (composed(2));
A) 6
B) 8
C) 10
D) 12
RESPOSTA: B

Difícil

1)Qual será o resultado da expressão?
const result = [1, 2, 3].map(x => x + 1).filter(x => x > 2).reduce((acc, x) => acc + x, 0)
A) 3
B) 4
C) 7
D) 8
RESPOSTA: C

2)Qual é a saída do seguinte código?
const compose = (...funcs) => x => funcs.reduceRight((acc, fn) => fn(acc), x);
const increment = x => x + 1;
*const square = x => x * x;*
const combined = compose(increment, square);
console.log(combined(3));
A) 9
B) 10
C) 11
D) 12
RESPOSTA: B

3)Qual é o resultado de

```
const f = x => x + 1
 const g = x => x * x
 const h = x => f(g(f(g(x))))
 console.log(h(2));?
```
A) 10
B) 11
C) 26
D) 27
RESPOSTA: C

4)Dado o seguinte código, qual será a saída?
const compose = (...funcs) => x => funcs.reduceRight((acc, fn) => fn(acc), x);
const add3 = x => x + 3;
*const multiply2 = x => x * 2;*
const combined = compose(add3, multiply2);
console.log(combined(5))
A) 8
B) 13
C) 10
D) 16
RESPOSTA: B

5)Qual das opções abaixo melhor descreve o resultado da execução do seguinte código?
const a = x => x + 2;
*const b = x => x * 3;*
const c = compose(a, b);
console.log(c(2) === 8);
A) Verdadeiro
B) Falso
C) Gera um erro
D) Retorna *undefined*
RESPOSTA: A

6)Considerando a função a seguir, qual será o resultado de *compose(f, g)(5)*?
const f = x => x > 10 ? x - 10 : x + 10;
*const g = x => x * 2;*
A) 10
B) 20
C) 30
D) 5
RESPOSTA: C

7)Qual é o resultado da execução do código abaixo?
const f = x => x + 3;
*const g = x => x * x;*
const h = x => f(g(x));
console.log(h(2) - h(3));

A) 0
B) -1
C) -3
D) -5
RESPOSTA: D

8)O que será impresso no console para a seguinte composição de funções?
const add = x => y => x + y;
const add5 = add(5);
const add10 = add(10);
console.log(add5(add10(5)));
A) 20
B) 15
C) 10
D) 5
RESPOSTA:A

9)O que acontece se você compuser uma função que espera um argumento com uma função que não aceita nenhum?
const f = () => 2;
const g = x => x + 1;
const h = compose(f, g);
console.log(h(5));
A) Retorna 2
B) Retorna 6
C) Gera um erro
D) Retorna 5
RESPOSTA: A

10)Qual é a saída do seguinte código?
*const f = x => x > 0 ? x * 2 : x - 1;*
const g = x => f(x + 1);
console.log(g(-2));
A) -1
B) 1
C) -2
D) 2
RESPOSTA: C