

Questões – Sobrecarga e Sobrescrita de Métodos em Java

Parte 1 – Questões objetivas

Questão 1 – Conceito de sobrecarga

Sobre a **sobrecarga de métodos (method overloading)** em Java, assinale a alternativa correta:

- A) Ocorre quando uma subclasse redefine um método herdado da superclasse.
- B) Ocorre quando uma classe possui dois ou mais métodos com o mesmo nome, mas com diferentes listas de parâmetros.
- C) Ocorre somente quando os métodos possuem diferentes tipos de retorno.
- D) Ocorre exclusivamente entre uma classe e sua superclasse.
- E) Ocorre quando dois métodos possuem exatamente a mesma assinatura.

Questão 2 – Identificando uma sobrecarga

Analise o código:

```
public class Calculadora {  
    public int somar(int a, int b) {  
        return a + b;  
    }  
  
    public double somar(double a, double b) {  
        return a + b;  
    }  
}
```

O código apresenta:

- A) Sobrescrita de métodos.
- B) Herança.
- C) Encapsulamento.
- D) Sobrecarga de métodos.
- E) Polimorfismo por herança.

Questão 3 – Parâmetros

Considere:

```
public class Exemplo {  
    public void imprimir(int valor) {  
        System.out.println(valor);  
    }  
  
    public void imprimir(String texto) {  
        System.out.println(texto);  
    }  
}
```

Por que esses dois métodos podem existir na mesma classe?

- A) Porque possuem tipos de retorno diferentes.
- B) Porque possuem nomes diferentes.
- C) Porque possuem listas de parâmetros diferentes.
- D) Porque são métodos `static`.
- E) Porque possuem modificadores de acesso diferentes.

Questão 4 – Tipo de retorno

Considere:

```
public class Teste {  
    public int calcular(int a, int b) {  
        return a + b;  
    }  
  
    public double calcular(int a, int b) {  
        return a + b;  
    }  
}
```

Esse código:

- A) É válido e representa sobrecarga.
- B) É válido e representa sobrescrita.
- C) Não compila porque métodos não podem ter o mesmo nome.
- D) Não compila porque apenas a alteração do tipo de retorno não caracteriza sobrecarga.
- E) Compila somente se um dos métodos for `static`.

Questão 5 – Conceito de sobrescrita

Sobre a **sobrescrita de métodos (method overriding)**, assinale a alternativa correta:

- A) Ocorre quando uma classe possui dois métodos com o mesmo nome e parâmetros diferentes.
- B) Ocorre quando uma subclasse fornece uma implementação própria para um método herdado da superclasse.
- C) Ocorre somente em métodos `private`.
- D) Ocorre somente entre métodos `static`.
- E) Ocorre quando dois métodos possuem tipos de retorno diferentes.

Questão 6 – Identificando uma sobrescrita

Analise:

```
class Animal {  
    public void emitirSom() {  
        System.out.println("Som do animal");  
    }  
}  
  
class Cachorro extends Animal {  
    @Override  
    public void emitirSom() {  
        System.out.println("Au au!");  
    }  
}
```

O código apresenta:

- A) Sobrecarga.
- B) Sobrescrita.
- C) Composição.
- D) Agregação.
- E) Associação.

Questão 7 – Anotação `@Override`

Qual é a principal finalidade da anotação `@Override`?

- A) Tornar o método privado.
- B) Informar ao compilador que o método pretende sobrescrever um método herdado.
- C) Permitir que o método seja sobrecarregado.
- D) Transformar o método em `static`.
- E) Impedir a herança da classe.

Questão 8 – Polimorfismo

Analise:

```
class Animal {  
    public void emitirSom() {  
        System.out.println("Som genérico");  
    }  
}  
  
class Cachorro extends Animal {  
    @Override  
    public void emitirSom() {  
        System.out.println("Au au!");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Animal animal = new Cachorro();  
        animal.emitirSom();  
    }  
}
```

Qual será a saída?

- A) Som genérico
- B) Au au!
- C) Erro de compilação.
- D) Nada será exibido.
- E) Animal Cachorro

Questão 9 – Sobrecarga e chamada do método

Considere:

```
class Calculadora {  
    public int multiplicar(int a, int b) {  
        return a * b;  
    }  
    public double multiplicar(double a, double b) {  
        return a * b;  
    }  
}
```

Ao executar:

```
Calculadora c = new Calculadora();  
System.out.println(c.multiplicar(2, 3));
```

```
System.out.println(c.multiplicar(2.5, 4.0));
```

Qual será a saída?

A)

6
10.0

B)

5
6.5

C)

6.0
10

D) Erro de compilação.

E) null e null.

Questão 10 – Diferença fundamental

Assinale a alternativa que apresenta corretamente a diferença entre **sobrecarga** e **sobrescrita**:

A) Sobrecarga ocorre entre classes diferentes e sobrescrita ocorre na mesma classe.

B) Sobrecarga utiliza métodos com mesmo nome e parâmetros diferentes; sobrescrita redefine um método herdado mantendo sua assinatura compatível.

C) Sobrecarga depende de herança; sobrescrita não depende de herança.

D) Sobrecarga e sobrescrita são exatamente o mesmo conceito.

E) Sobrescrita ocorre quando apenas o tipo de retorno é alterado.

Parte 2 – Questões discursivas

Questão 11

Explique, com suas palavras, o que é **sobrecarga de métodos** em Java.

Apresente um exemplo de uma classe contendo dois métodos sobrecarregados.

Questão 12

Explique o conceito de **sobrescrita de métodos** em Java.

Utilize uma superclasse e uma subclasse para exemplificar.

Questão 13

Analise o código:

```

class Pessoa {
    public void apresentar() {
        System.out.println("Sou uma pessoa.");
    }
}

class Professor extends Pessoa {
    @Override
    public void apresentar() {
        System.out.println("Sou um professor.");
    }
}

```

Responda:

- a)** Qual conceito de POO está sendo utilizado?
- b)** Qual método será executado no código abaixo?

```

Pessoa pessoa = new Professor();
pessoa.apresentar();

```

- c)** Explique por que esse método é executado.

Questão 14

Observe:

```

class Calculadora {
    public int somar(int a, int b) {
        return a + b;
    }

    public int somar(int a, int b, int c) {
        return a + b + c;
    }

    public double somar(double a, double b) {
        return a + b;
    }
}

```

Responda:

- a)** Quantos métodos `somar ( )` existem na classe?
- b)** O código apresenta sobrecarga ou sobrescrita?
- c)** Explique quais características permitem que esses métodos tenham o mesmo nome.

Questão 15

Considere:

```
class Veiculo {  
    public void acelerar() {  
        System.out.println("Veículo acelerando");  
    }  
}  
  
class Carro extends Veiculo {  
    @Override  
    public void acelerar() {  
        System.out.println("Carro acelerando");  
    }  
}
```

Crie um programa `main` que demonstre o **polimorfismo** utilizando essas classes.

Gabarito

Questões objetivas

Questão	Resposta	Justificativa
1	B	Sobrecarga ocorre quando métodos possuem o mesmo nome, mas listas de parâmetros diferentes.
2	D	Os métodos <code>somar ()</code> possuem o mesmo nome, mas parâmetros de tipos diferentes.
3	C	A lista de parâmetros é diferente: <code>int</code> e <code>String</code> .
4	D	Alterar somente o tipo de retorno não caracteriza sobrecarga.
5	B	Sobrescrita ocorre quando uma subclasse redefine um método herdado.
6	B	<code>Cachorro</code> redefine o método <code>emitirSom ()</code> de <code>Animal</code> .
7	B	<code>@Override</code> informa a intenção de sobrescrever um método herdado e permite ao compilador verificar isso.
8	B	Apesar da referência ser <code>Animal</code> , o objeto é um <code>Cachorro</code> ; ocorre polimorfismo em tempo de execução.
9	A	O Java seleciona o método adequado de acordo com os argumentos fornecidos.
10	B	Essa é a distinção fundamental entre sobrecarga e sobrescrita.

Respostas das questões discursivas

Questão 11

Resposta esperada:

Sobrecarga ocorre quando uma classe possui dois ou mais métodos com o **mesmo nome**, mas com **diferentes listas de parâmetros**.

Exemplo:

```
class Calculadora {  
    public int somar(int a, int b) {  
        return a + b;  
    }  
  
    public int somar(int a, int b, int c) {  
        return a + b + c;  
    }  
}
```

Os métodos possuem o mesmo nome `somar`, mas quantidades diferentes de parâmetros.

Questão 12

Resposta esperada:

Sobrescrita ocorre quando uma subclasse fornece uma nova implementação para um método herdado de uma superclasse.

Exemplo:

```
class Animal {  
    public void emitirSom() {  
        System.out.println("Som do animal");  
    }  
}  
  
class Cachorro extends Animal {  
    @Override  
    public void emitirSom() {  
        System.out.println("Au au!");  
    }  
}
```

Nesse caso, `Cachorro` sobrescreve o método `emitirSom()` definido em `Animal`.

Questão 13

Resposta:

a) O código utiliza **herança, sobrescrita e polimorfismo**.

b) A saída será:

Sou um professor.

c) Embora a variável seja declarada como:

Pessoa pessoa

o objeto criado é:

new Professor()

Durante a execução, o Java identifica que o objeto é um **Professor** e executa a implementação sobrescrita de **apresentar()**.

Questão 14

Resposta:

a) Existem **três métodos** **somar()**.

b) O código apresenta **sobrecarga**.

c) Os métodos possuem o mesmo nome, mas possuem diferentes listas de parâmetros:

```
somar(int, int)
somar(int, int, int)
somar(double, double)
```

A diferença na lista de parâmetros permite ao Java identificar qual método deve ser chamado.

Questão 15

Resposta esperada:

```
public class Main {
    public static void main(String[] args) {
        Veiculo veiculo = new Carro();
        veiculo.acelerar();
    }
}
```

Saída:

Carro acelerando

Nesse exemplo:

```
Veiculo veiculo = new Carro();
```

a variável de referência é do tipo **Veiculo**, mas o objeto é um **Carro**.

Quando:

```
veiculo.acelerar();
```

é executado, o Java utiliza a implementação sobrescrita de `Carro`. Isso demonstra **polimorfismo em tempo de execução**.