

Questões — new e Construtores em Java

Questão 1 — Múltipla escolha

Considere o código:

```
public class Pessoa {  
    String nome;  
}
```

Para criar um objeto da classe `Pessoa`, qual instrução está correta?

- A) `Pessoa p;`
 - B) `Pessoa p = Pessoa();`
 - C) `Pessoa p = new Pessoa();`
 - D) `new Pessoa p;`
 - E) `Pessoa.new p();`
-

Questão 2 — Múltipla escolha

Qual é a principal finalidade da palavra-chave `new` em Java?

- A) Declarar uma classe.
 - B) Criar uma nova instância de uma classe.
 - C) Declarar um método.
 - D) Definir uma variável como constante.
 - E) Fazer a herança entre duas classes.
-

Questão 3 — Múltipla escolha

Analise o código:

```
public class Produto {  
    private String nome;  
  
    public Produto() {  
        nome = "Produto padrão";  
    }  
}
```

Ao executar:

```
Produto p = new Produto();
```

o que acontece?

- A) A variável `p` recebe `null`.
- B) É criado um objeto `Produto` e seu construtor é executado.

- C) O método `Produto()` precisa ser chamado separadamente.
 - D) O código apresenta erro porque construtores não podem ser chamados com `new`.
 - E) O atributo `nome` recebe automaticamente `null`.
-

Questão 4 — Múltipla escolha

Sobre os construtores em Java, assinale a alternativa **correta**.

- A) Um construtor deve obrigatoriamente possuir retorno `void`.
 - B) O construtor possui obrigatoriamente o mesmo nome da classe.
 - C) Um construtor pode possuir qualquer nome.
 - D) Construtores são chamados somente por meio de métodos `static`.
 - E) Um construtor deve sempre ser declarado como `private`.
-

Questão 5 — Múltipla escolha

Considere:

```
public class Aluno {  
  
    private String nome;  
    private int idade;  
  
    public Aluno(String nome, int idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Qual instrução cria corretamente um objeto dessa classe?

- A) `Aluno aluno = new Aluno();`
 - B) `Aluno aluno = Aluno("João", 20);`
 - C) `Aluno aluno = new Aluno("João", 20);`
 - D) `new Aluno aluno("João", 20);`
 - E) `Aluno aluno("João", 20);`
-

Questão 6 — Múltipla escolha

Observe:

```
public class Carro {  
  
    private String modelo;
```

```
public Carro() {  
    modelo = "Não informado";  
}  
  
public Carro(String modelo) {  
    this.modelo = modelo;  
}  
}
```

Qual conceito está sendo utilizado nesse exemplo?

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

Questão 7 — Verdadeiro ou falso

Analise as afirmações:

- I. A palavra-chave `new` pode ser utilizada para criar uma instância de uma classe.
- II. Um construtor possui o mesmo nome da classe.
- III. Um construtor possui obrigatoriamente um tipo de retorno.
- IV. O construtor é executado quando um objeto é criado utilizando `new`.

Assinale a alternativa correta:

- A) V — V — F — V
- B) V — F — F — V
- C) F — V — V — F
- D) V — V — V — V
- E) F — F — F — V

Questão 8 — Analise o código

```
public class Conta {  
  
    private double saldo;  
  
    public Conta(double saldoInicial) {  
        saldo = saldoInicial;  
    }  
}
```

Considere:

```
Conta conta = new Conta(1500.00);
```

Responda:

- Qual é a função da palavra-chave `new` nesse código?
 - Qual construtor está sendo chamado?
 - Qual será o valor inicial do atributo `saldo`?
-

Questão 9 — Questão discursiva

Explique a diferença entre **declarar uma variável de referência** e **criar um objeto** em Java.

Utilize como exemplo:

```
Pessoa pessoa;
```

e

```
Pessoa pessoa = new Pessoa();
```

Questão 10 — Questão discursiva

Observe:

```
public class Funcionario {  
  
    private String nome;  
    private double salario;  
  
    public Funcionario(String nome, double salario) {  
        this.nome = nome;  
        this.salario = salario;  
    }  
}
```

Escreva uma instrução Java que crie um objeto `Funcionario` com:

- Nome: "Carlos"
 - Salário: 3500.00
-

Questão 11 — Analise o erro

Considere a classe:

```
public class Livro {
```

```
private String titulo;

public Livro(String titulo) {
    this.titulo = titulo;
}
}
```

Um programador escreveu:

```
Livro livro = new Livro();
```

Essa instrução compila? Explique o motivo.

Questão 12 — Construtor padrão

Considere:

```
public class Pessoa {

    private String nome;
}
```

O código abaixo é válido?

```
Pessoa pessoa = new Pessoa();
```

Explique o que acontece em relação ao construtor da classe.

Gabarito

Questão 1

Resposta: C

```
Pessoa p = new Pessoa();
```

A declaração cria uma referência **p** e o **new** cria uma nova instância de **Pessoa**.

Questão 2

Resposta: B

A palavra-chave **new** é utilizada para **criar uma nova instância de uma classe**, alocando o objeto e acionando seu construtor.

Questão 3

Resposta: B

O **new** cria uma instância de **Produto** e, durante a criação, o construtor:

```
public Produto()
```

é executado.

Consequentemente:

```
nome = "Produto padrão";
```

é executado.

Questão 4

Resposta: B

O construtor deve possuir o **mesmo nome da classe**.

Exemplo:

```
public class Pessoa {  
  
    public Pessoa() {  
    }  
}
```

Observe que o construtor `Pessoa()` não possui tipo de retorno.

Questão 5

Resposta: C

```
Aluno aluno = new Aluno("João", 20);
```

O construtor exige dois parâmetros:

```
Aluno(String nome, int idade)
```

Questão 6

Resposta: C — Sobrecarga de construtores

A classe possui dois construtores:

```
public Carro()
```

e

```
public Carro(String modelo)
```

Eles possuem o mesmo nome, mas listas de parâmetros diferentes.

Questão 7

Resposta: A

V — V — F — V

- I — Verdadeiro.
 - II — Verdadeiro.
 - III — Falso. Construtores **não possuem tipo de retorno**.
 - IV — Verdadeiro.
-

Questão 8

a) A palavra-chave `new` cria uma nova instância da classe `Conta`.

b) É chamado o construtor:

```
public Conta(double saldoInicial)
```

c) O valor inicial de `saldo` será:

```
1500.00
```

Questão 9

`Pessoa pessoa;` apenas **declara uma variável de referência**. Nesse momento, nenhum objeto `Pessoa` foi criado.

Já:

```
Pessoa pessoa = new Pessoa();
```

faz duas coisas:

1. Declara a referência `pessoa`.
 2. Cria uma nova instância de `Pessoa` utilizando `new`.
-

Questão 10

Resposta esperada:

```
Funcionario funcionario = new Funcionario("Carlos", 3500.00);
```

O `new` cria o objeto e chama o construtor correspondente.

Questão 11

Não compila.

A classe possui somente o construtor:

```
public Livro(String titulo)
```

Portanto, não existe um construtor sem parâmetros:

```
Livro()
```

Assim, a instrução:

```
Livro livro = new Livro();
```

gera erro de compilação.

Questão 12

Sim, é válido.

Quando uma classe **não declara nenhum construtor**, o Java fornece automaticamente um **construtor padrão sem parâmetros**.

Assim:

```
public class Pessoa {  
    private String nome;  
}
```

permite:

```
Pessoa pessoa = new Pessoa();
```

Porém, se a classe declarar explicitamente um construtor, o construtor padrão não será criado automaticamente.

Por exemplo:

```
public class Pessoa {  
  
    private String nome;  
  
    public Pessoa(String nome) {  
        this.nome = nome;  
    }  
}
```

Nesse caso:

```
Pessoa pessoa = new Pessoa();
```

não compila, pois não existe construtor sem argumentos.