

Lista de revisão - Revisão segunda fase

1. Considere o código abaixo

```
class Funcionario {  
    public void trabalhar() {  
        System.out.println("Trabalhando...");  
    }  
}  
class Gerente extends Funcionario { }
```

Ao executar:

```
Gerente g = new Gerente();  
g.trabalhar();
```

Qual será o resultado?

- a) Erro de compilação
- b) Impressão: "Trabalhando..."
- c) Nenhuma saída
- d) Impressão: "Gerente trabalhando..."

2. Para que a classe **Aluno** herde de uma classe **Pessoa**, qual das opções abaixo é usada?

- a) class Aluno inherits Pessoa
- b) class Aluno extends Pessoa
- c) class Pessoa extends Aluno
- d) class Aluno implements Pessoa

3. O que acontece ao usar `super()` dentro de um construtor de uma subclasse?

- a) Chama o construtor da própria classe.
- b) Chama o construtor da superclasse.
- c) Chama o método estático da superclasse.
- d) Chama o método sobrescrito da subclasse.

4. Suponha o seguinte código:

```
Map<String, Integer> estoque = new HashMap<>();  
estoque.put("Caneta", 50);  
estoque.put("Lápis", 30);
```

Qual método deve ser usado para obter a quantidade de "Caneta"?

- a) `estoque.buscar("Caneta")`
- b) `estoque.get("Caneta")`
- c) `estoque.find("Caneta")`
- d) `estoque.findByKey("Caneta")`

5. Considere:

```
List<String> nomes = new ArrayList<>();  
nomes.add("Ana");  
nomes.add("João");  
nomes.add("Carlos");
```

Qual método retorna o número total de nomes?

- A) `nomes.count()`
- B) `nomes.total()`
- C) `nomes.size()`
- D) `nomes.length()`

6. Sobre o relacionamento entre classes e objetos em Java, marque a alternativa correta:

- a) Um objeto define a estrutura de uma classe.
- b) Uma classe é um exemplo concreto de um objeto.
- c) Um objeto é uma instância de uma classe.
- d) Uma classe é sempre estática.

7. O que diferencia um **construtor** de um método comum em Java?

- a) Ele sempre retorna um valor.
- b) Seu nome deve ser diferente do nome da classe.
- c) Ele é invocado automaticamente na criação de um objeto.
- d) Ele só pode ser usado em classes abstratas.

8. Considere:

```
abstract class Documento {  
    public abstract void imprimir();  
    public void salvar() {  
        System.out.println("Salvando documento...");  
    }  
}
```

Qual afirmativa é correta?

- a) A classe Documento pode ser instanciada.
- b) Toda classe que herdar de Documento deve implementar o método salvar().
- c) O método salvar() precisa ser abstrato.
- d) A classe Documento define comportamento comum e obrigatório para suas subclasses. O método salvar() precisa ser abstrato.

9. O que representa o uso da abstração na POO?

- a) Esconder a lógica de execução e mostrar apenas o que é necessário.
- b) Evitar o uso de métodos públicos.
- c) Criar classes que contenham apenas atributos.
- d) Garantir que uma classe não seja herdada.

10. Considere:

```
class Produto {  
    private double preco;  
    public void setPreco(double p) {  
        if (p > 0) this.preco = p;  
    }  
    public double getPreco() {  
        return preco;  
    }  
}
```

Por que o código acima representa o princípio de encapsulamento?

- a) Os atributos são acessados diretamente.
- b) O uso de modificadores privados evita que o código funcione.
- c) O acesso ao dado é feito por métodos controlados.
- d) O método setPreco impede a visibilidade do dado.

11. Por que é recomendado declarar atributos como private em uma classe?

- a) Para que eles sejam compartilhados com outras classes.
- b) Para que possam ser acessados apenas dentro da própria classe.
- c) Para que tenham visibilidade global.
- d) Para impedir que eles sejam inicializados.

12. Dado:

```
class Pessoa {  
    public String nome;  
}  
class Estudante extends Pessoa {  
    public String matricula;  
}
```

Qual alternativa é verdadeira?

- a) A classe Pessoa herda de Estudante.
- b) A classe Estudante herda os atributos da classe Pessoa.
- c) Estudante não pode acessar o atributo nome.
- d) A herança só funciona com interfaces.

13. Dado o enum:

```
public enum Prioridade {  
    BAIXA, MEDIA, ALTA  
}
```

Qual comando retorna todas as constantes definidas?

- a) Prioridade.values()
- b) Prioridade.all()
- c) Prioridade.getList()
- d) Prioridade.toArray()

14. Considere:

```
enum Status {  
    ATIVO(1), INATIVO(0);  
    private int codigo;  
    Status(int codigo) {  
        this.codigo = codigo;  
    }  
}
```

```
        public int getCodigo() {  
            return codigo;  
        }  
    }  
}
```

O que será impresso ao executar:

```
System.out.println(Status.ATIVO.getCodigo());
```

- a) ATIVO
- b) 1
- c) 0
- d) Erro de compilação

15. Qual das alternativas descreve corretamente o uso de enum em uma estrutura de repetição?

- a) Enums não podem ser percorridos.
- b) É possível iterar sobre os valores de um enum com um for usando values().
- c) Só é possível percorrer enums com um switch.
- d) Enums só funcionam dentro de métodos estáticos.

GABARITO

1 – B

2 – B

3 – B

4- B

5 – A

6 - C

7 – C

8- D

9 – A

10 – C

11 – B

12 – B

13 - A

14 – B

15 - B