**Simulação de Prova - Estruturas de Dados**

**Instruções:** Leia atentamente cada questão e responda o que se pede.

**Seção 1: Questões de Múltipla Escolha**

*(Instrução: Marque a alternativa que melhor responde à questão)*

1. **Analise o trecho de código Java do arquivo AULA 2 - Codigo que popula uma lista com n numeros aleatorios inteiros.java:**

Java

1if (!lista.contains(numero)) {  
2 lista.add(numero);  
3}

**Qual a finalidade principal do método contains() neste contexto?**

a) Ordenar os números na lista.  
b) Verificar se um número já existe na lista antes de adicioná-lo, evitando duplicatas.  
c) Remover um número específico da lista.  
d) Contar quantos elementos existem na lista.

**Resposta e Explicação:**

**Resposta Correta:** b) Verificar se um número já existe na lista antes de adicioná-lo, evitando duplicatas.

**Explicação:** O método **lista.contains(numero)** retorna true se o número já estiver presente. A condição **if (!lista.contains(numero))** garante que apenas números únicos sejam adicionados.

1. **No arquivo AULA 1 - INTRODUÇÃO ARQUIVO EM C .c, qual o propósito das variáveis vetorP e vetorI?**

C

1int vetor[10], \*vetorP[10], \*vetorI[10], i, tamP = 0, tamI = 0;   
2// ...  
3if(vetor[i] % 2 == 0) {  
4 vetorP[tamP++] = vetor[i]; // Linha relevante   
5} else {  
6 vetorI[tamI++] = vetor[i]; // Linha relevante   
7}

a) Armazenar todos os números do vetor original.  
b) Armazenar ponteiros para os elementos pares (vetorP) e ímpares (vetorI) do vetor original.  
c) vetorP armazena os índices pares e vetorI os índices ímpares.  
d) São usados para calcular a média dos números pares e ímpares.

**Resposta e Explicação:**

**Resposta Correta:** b) Armazenar ponteiros para os elementos pares (vetorP) e ímpares (vetorI) do vetor original.

**Explicação:** As variáveis **vetorP** e **vetorI** são arrays de ponteiros que tentam separar os números do vetor original com base na paridade.

1. **Com base nas explicações em AULA 2 - Tarefas Da Aula.java, qual a principal diferença entre ArrayList e Vector em Java?**

a) ArrayList é mais rápido para inserção e remoção, enquanto Vector é mais rápido para busca.  
b) Vector não pode crescer dinamicamente, ao contrário do ArrayList.  
c) Vector é sincronizado (thread-safe) e geralmente mais lento, enquanto ArrayList não é sincronizado.  
d) ArrayList só armazena tipos primitivos, enquanto Vector armazena objetos.

**Resposta e Explicação:**

**Resposta Correta:** c) Vector é sincronizado (thread-safe) e geralmente mais lento, enquanto ArrayList não é sincronizado.

**Explicação:** Vector é seguro para uso concorrente, mas mais lento. ArrayList é mais rápido em ambientes de thread única.

1. **Qual estrutura de dados é utilizada no arquivo AULA 4 - DICIONARIOS.java para associar nomes (chaves) a idades (valores)?**

Java

1Map<String, Integer> dicionario1 = new HashMap<>();  
2dicionario1.put("João", 25);

a) ArrayList  
b) LinkedList  
c) Vector  
d) HashMap

**Resposta e Explicação:**

**Resposta Correta:** d) HashMap

**Explicação:** **HashMap** é uma implementação de **Map** que armazena pares chave-valor, permitindo associações entre nomes e idades.

1. **No código da classe Aluno (presente nos arquivos da AULA 6 em Java, C# e Python), qual a importância de sobrescrever/implementar os métodos equals (Equals/eq) e hashCode (GetHashCode/hash)?**

a) Para permitir a ordenação da lista de alunos pelo nome.  
b) Para gerar o email do aluno automaticamente.  
c) Para definir como comparar dois objetos Aluno e garantir o funcionamento em coleções como HashMap.  
d) Para converter o objeto Aluno em uma representação String.

**Resposta e Explicação:**

**Resposta Correta:** c) Para definir como comparar dois objetos Aluno e garantir o funcionamento em coleções como HashMap.

**Explicação:** O método **equals** define a igualdade entre objetos, enquanto **hashCode** deve ser consistente com **equals** para funcionar corretamente em coleções baseadas em hash.

1. **O código em AULA 4 - EXERCICIO CONTADOR DISCIPLINAS.java tem como objetivo principal:**

a) Cadastrar alunos em diferentes disciplinas.  
b) Calcular a média das notas dos alunos em cada disciplina.  
c) Listar todos os alunos matriculados no curso de Ciência da Computação.  
d) Contar em quantas disciplinas cada aluno está matriculado, percorrendo as listas de cada disciplina.

**Resposta e Explicação:**

**Resposta Correta:** d) Contar em quantas disciplinas cada aluno está matriculado.

**Explicação:** O código itera sobre uma lista de alunos e verifica em quantas disciplinas cada um está matriculado, contando as ocorrências.

**Seção 2: Questões Discursivas**

1. **Explique a diferença fundamental entre ArrayList e LinkedList em Java.**

**Resposta e Explicação:**

**ArrayList:**

* 1. Usa um array dinâmico.
  2. Acesso por índice é rápido (O(1)).
  3. Inserções/remoções no meio são lentas (O(n)).

**LinkedList:**

* 1. Usa uma lista duplamente encadeada.
  2. Acesso por índice é lento (O(n)).
  3. Inserções/remoções no meio são rápidas (O(1)).

**Conclusão:** Use **ArrayList** para acesso rápido por índice e **LinkedList** para inserções/remoções frequentes.

1. **Descreva o que o método gerarEmail na classe Aluno faz e como trata emails duplicados.**

**Resposta e Explicação:**

O método **gerarEmail** cria um e-mail padrão a partir do nome do aluno. Se o e-mail já existir, ele gera variações adicionando números até encontrar um único. Isso garante que cada aluno tenha um e-mail exclusivo.

**Seção 3: Questões de Programação**

1. **(Java) Escreva um método estático em Java chamado filtrarPares que receba um ArrayList<Integer> e retorne um novo ArrayList<Integer> com apenas os números pares.**

**Resposta e Explicação (Código Java):**

Java

1import java.util.ArrayList;  
2  
3public class FiltroNumeros {  
4 public static ArrayList<Integer> filtrarPares(ArrayList<Integer> original) {  
5 ArrayList<Integer> pares = new ArrayList<>();  
6 if (original == null) return pares;  
7  
8 for (Integer numero : original) {  
9 if (numero != null && numero % 2 == 0) {  
10 pares.add(numero);  
11 }  
12 }  
13 return pares;  
14 }  
15  
16 public static void main(String[] args) {  
17 ArrayList<Integer> numeros = new ArrayList<>();  
18 numeros.add(1);  
19 numeros.add(2);  
20 numeros.add(3);  
21 numeros.add(4);  
22 numeros.add(5);  
23 numeros.add(6);  
24 numeros.add(null);  
25 numeros.add(8);  
26 numeros.add(-10);  
27  
28 System.out.println("Lista original: " + numeros);  
29 ArrayList<Integer> listaPares = filtrarPares(numeros);  
30 System.out.println("Lista de pares filtrada: " + listaPares);  
31 }  
32}

**Explicação da Lógica:** O método percorre a lista original, verifica se os números são pares e não nulos, e os adiciona a uma nova lista, que é retornada.

1. **(Python) Crie uma classe chamada Produto com atributos codigo e nome. Escreva um código que crie uma lista para armazenar objetos Produto.**

**Resposta e Explicação (Código Python):**

Python

1class Produto:  
2 def \_\_init\_\_(self, codigo: int, nome: str):  
3 self.codigo = codigo  
4 self.nome = nome  
5  
6 def \_\_str\_\_(self):  
7 return f"Produto(Código: {self.codigo}, Nome: '{self.nome}')"  
8  
9if \_\_name\_\_ == "\_\_main\_\_":  
10 lista\_produtos = []  
11 produto1 = Produto(101, "Teclado Mecânico")  
12 produto2 = Produto(102, "Mouse Óptico")  
13 produto3 = Produto(205, "Monitor LED 27\"")  
14  
15 lista\_produtos.append(produto1)  
16 lista\_produtos.append(produto2)  
17 lista\_produtos.append(produto3)  
18  
19 for i, item in enumerate(lista\_produtos):  
20 print(f"Item {i+1}: {item}")

**Explicação da Lógica:** A classe **Produto** armazena código e nome. No bloco principal, três instâncias são criadas e adicionadas a uma lista, que é percorrida para imprimir os detalhes.