



CONCURSO

○ PROFESSOR EFETIVO

. CEFET-MG .

2024



DOCENTE EBTT CAMPUS BELO HORIZONTE

EDITAL ESPECÍFICO Nº 07/2024

ETAPA DA PROVA ESCRITA

ENGENHARIA DE COMPUTAÇÃO

INSTRUÇÕES

1. Confira se as informações sobre a Área do Concurso, o número do Edital e o Campus, que estão descritas na capa deste Caderno de Prova, estão de acordo com os dados do seu comprovante definitivo de inscrição.
2. Inicie a prova lendo as instruções que estão no Caderno de Prova.
3. A ordem de resolução das questões é uma escolha do próprio candidato, porém elas devem ser respondidas na folha definitiva de respostas, obrigatoriamente, na ordem em que constam no Caderno de Prova.
4. **Atenção:** As folhas de rascunho não serão corrigidas pela Banca Examinadora. As folhas definitivas de respostas são os únicos instrumentos que serão avaliados e sob nenhuma hipótese serão substituídas.
5. É proibido assinar e fazer qualquer tipo de marcação no Caderno de Prova, nas folhas definitivas de respostas e nas folhas de rascunho, sob pena de eliminação deste Concurso Público.
6. Responda às questões nas folhas definitivas de respostas fornecidas pelo aplicador.
7. Nenhuma folha deverá ser destacada do Caderno de Prova, das folhas definitivas de respostas ou do rascunho.
8. Não serão fornecidas folhas extras para rascunho ou para responder às questões de prova.
9. O Caderno de Prova, as folhas definitivas de respostas e as folhas de rascunho deverão ser devolvidos juntos para o Aplicador.
10. O tempo regulamentar de prova é de 4:30 (quatro horas e trinta minutos) e será exigido um tempo mínimo de permanência em sala de 1 (uma) hora, para garantir o sigilo da prova.
11. Esta prova contém 06 (seis) questões (03 objetivas e 03 discursivas) e será avaliada em 100,00 (cem pontos).
12. Durante a realização da Prova Escrita, o candidato poderá portar somente: caneta esferográfica de tinta preta ou azul, de corpo transparente.

QUESTÃO 01 (10 pontos)

Considere os algoritmos de Bellman-Ford, Dijkstra e Floyd-Warshall, que solucionam problemas do caminho mínimo em um grafo $G(V, A)$, direcionado e ponderado, em que V é o conjunto de vértices e A é o conjunto de arestas. Sobre eles, analise se as afirmativas abaixo são verdadeiras (V) ou falsas (F).

- () A implementação do algoritmo de Dijkstra com matriz de adjacência possui uma complexidade computacional $O(|V|^3)$.
- () O paradigma de programação usado nas implementações de Bellman-Ford, Dijkstra e Floyd-Warshall é a programação dinâmica.
- () A implementação do algoritmo de Dijkstra, que adota fila de prioridade, implementada como heap de Fibonacci, possui uma complexidade computacional $O(|V| \log |V| + |A|)$.
- () O melhor caso do algoritmo Floyd-Warshall possui complexidade computacional de $O((|V| + |A|) \log |V|)$, que é quando não há a necessidade de relaxar muitas arestas.
- () O algoritmo de Floyd-Warshall consegue resolver o problema do caminho mínimo quando existem arestas de peso negativo, desde que não existam ciclos de peso negativo.

A sequência correta é

- a) F, F, F, V, V.
- b) F, F, V, F, V.
- c) F, V, V, F, F.
- d) V, F, V, F, F.
- e) V, V, F, F, V.

QUESTÃO 02 (10 pontos)

Em Programação Orientada a Objetos (POO), o polimorfismo é um conceito fundamental que permite que diferentes classes derivadas implementem comportamentos específicos, mantendo a mesma interface definida na classe base. Sobre o polimorfismo, é correto afirmar que ele

- a) é garantido quando os métodos são declarados explicitamente na classe derivada, sem a necessidade de qualquer indicação na classe base.
- b) pode ser aplicado somente quando os métodos são sobrecarregados (*overloaded*), ou seja, possuem o mesmo nome, mas diferentes assinaturas.
- c) permite que o método a ser executado seja determinado pelo compilador, com base no tipo da variável, em tempo de compilação.
- d) possibilita que objetos de classes derivadas sejam tratados como objetos da classe base, mas os métodos específicos da classe derivada só podem ser chamados diretamente por meio de uma referência da classe derivada.
- e) permite que métodos das classes derivadas sejam chamados por meio de referências ou ponteiros da classe base, desde que esses métodos sejam definidos de forma adequada na classe base para permitir essa flexibilidade.

QUESTÃO 03 (10 pontos)

Sobre o processo unificado de desenvolvimento de software, avalie se as afirmações abaixo são verdadeiras (V) ou falsas (F):

- () O processo unificado de desenvolvimento de software é o conjunto de atividades necessárias para transformar os requisitos do usuário em um sistema de software baseado em componentes.
- () Os aspectos que distinguem o processo unificado são capturados em três conceitos chaves: direcionado a casos de uso; iterativo e incremental; centrado em dados.
- () Uma das características do processo unificado é ser centrado na arquitetura. Isso contribui para organizar o esforço de desenvolvimento, dividindo o software em porções discretas, permitindo, assim, o reúso de componentes dentro dessas porções.
- () As quatro fases do processo unificado são: concepção, elaboração, construção e transição. Essas fases contêm iterações sequenciais, que por sua vez, podem conter os cinco fluxos de trabalho: requisito, análise, projeto, implementação e teste.

A sequência correta é

- a) F, V, F, F.
- b) F, V, V, V.
- c) V, F, V, V.
- d) V, V, F, F.
- e) V, F, V, F.

QUESTÃO 04 (25 pontos)

Considere a estrutura de dados Árvore de Busca Binária, implementada em C/C++ com uma estrutura que representa um nó o qual armazena um valor inteiro como chave e dois ponteiros que o ligam, respectivamente, aos nós da esquerda e da direita.

- a) APRESENTE um trecho de código-fonte que define a estrutura de dados.
- b) Para as operações de inserção, remoção e busca de um elemento na árvore, (I) EXPLIQUE os passos a serem executados, (II) INFORME o custo computacional no melhor e pior caso e (III) EXPLIQUE quando ocorrem tais casos.

QUESTÃO 05 (23 pontos)

Em sistemas concorrentes, o impasse (*deadlock*) é uma preocupação significativa que pode comprometer a confiabilidade e a performance do sistema.

- a) EXPLIQUE o conceito de impasse (*deadlock*) em sistemas concorrentes, identificando as condições necessárias para que ele ocorra. IDENTIFIQUE os riscos associados ao impasse em sistemas de computação.
- b) EXPLIQUE, por meio de um exemplo prático, como a ausência de uma sincronização adequada entre tarefas pode levar à ocorrência de impasses, especialmente em cenários em que várias tarefas competem por recursos limitados.
- c) ESCREVA uma solução baseada na técnica de passagem de mensagens para evitar impasses em sistemas concorrentes. EXPLIQUE como essa técnica pode ser utilizada para garantir que as tarefas compartilhem recursos sem provocar condições de impasse.

QUESTÃO 06 (22 pontos)

A Programação extrema (XP) é considerada uma metodologia ágil, pois se ajusta bem a pequenas e médias equipes de desenvolvimento de software com requisitos vagos e em constante mudança. A Integração Contínua (CI) é uma prática que foi introduzida pela XP.

- a) DEFINA CI e DESCREVA sobre a sua finalidade. DISCORRA sobre três boas práticas para o uso de CI.
- b) CITE três exemplos de servidores de integração contínua adotados nos projetos de software.