

Questionário Capítulo 1

Total de pontos 6/10 ?

E-mail *

martinson.freitas@gmail.com

✓ O que são estruturas de dados e por que são importantes? *

1/1

- ☐ São programas que protegem contra vírus
- ☐ São ferramentas para criar interfaces gráficas
- ☒ São formas de armazenar e organizar dados em um programa de computador, importantes para a eficiência e funcionalidade do código. ✓
- ☐ São técnicas para criptografar informações confidenciais
- ☐ São métodos para criar backups de arquivos



- ☒ A forma como a estrutura de dados é implementada ✗
- ☐ A documentação que descreve como usar a estrutura de dados
- ☐ O conjunto de operações que podem ser realizadas na estrutura de dados
- ☐ O nome dado à variável que armazena a estrutura de dados
- ☐ Nenhuma das opções acima

Resposta correta

- ☒ O conjunto de operações que podem ser realizadas na estrutura de dados

Feedback

Uma interface, às vezes também chamada de tipo abstrato de dados, define o conjunto de operações suportado por uma estrutura de dados e a semântica, ou significado, dessas operações. Uma interface não nos diz nada sobre como a estrutura de dados implementa essas operações; ela fornece somente uma lista de operações suportadas junto com especificações sobre quais tipos de argumentos cada operação aceita e o valor retornado por cada operação. Ao discutir estruturas de dados, é importante entender a diferença entre a interface de uma estrutura de dados e sua implementação. Uma interface descreve o que uma estrutura de dados faz, enquanto uma implementação descreve como a estrutura de dados o faz.

✓ Quantas operações por segundo um computador desktop muito rápido pode fazer? *1/1

- ☐ 1 milhão (10^6)
- ☒ 1 bilhão (10^9) ✓
- ☐ 1 trilhão (10^{12})
- ☐ 1 quatrilhão (10^{15})
- ☐ Nenhuma das opções acima

✓ Qual é a diferença entre uma interface e uma implementação em relação a uma estrutura de dados? *1/1

a)

- ☒ Uma interface descreve o que uma estrutura de dados faz, enquanto uma implementação descreve como a estrutura de dados faz. ✓
- ☐ Uma interface descreve como a estrutura de dados faz, enquanto uma implementação descreve o que a estrutura de dados faz.
- ☐ Uma interface e uma implementação são sinônimos e podem ser usados indistintamente.
- ☐ Uma interface é usada apenas em linguagens de programação orientadas a objetos, enquanto uma implementação é usada em linguagens procedurais.
- ☐ Nenhuma das opções acima

✓ Quanto tempo levaria para realizar 1 trilhão de inspeções com um computador desktop muito rápido? *1/1

- ☐ Cerca de 1 segundo
- ☒ Cerca de 16 minutos e 40 segundos ✓
- ☐ Cerca de 1 hora
- ☐ Cerca de 24 horas
- ☐ Nenhuma das opções acima



✗ Como as estruturas de dados do sistema de arquivos são usadas para abrir um arquivo? *.../1

- ☐ Para criptografar o conteúdo do arquivo
- ☒ Para localizar as partes do arquivo no disco para que possam ser recuperadas ✗
- ☐ Para compactar o tamanho do arquivo
- ☐ Para criar backups automáticos do arquivo
- ☐ Nenhuma das opções acima

Nenhuma resposta correta

✗ Qual é a importância da compreensão das interfaces e implementações das estruturas de dados na programação? *0/1

- ☐ Ajuda a melhorar a eficiência do código
- ☐ Ajuda a evitar erros comuns na programação
- ☒ Permite projetar e implementar estruturas de dados de forma mais efetiva ✗
- ☐ Todas as opções acima
- ☐ Nenhuma das opções acima

Resposta correta

- ☒ Ajuda a melhorar a eficiência do código

Feedback

Compreender as interfaces e implementações das estruturas de dados é importante na programação porque uma interface descreve o que uma estrutura de dados faz, enquanto uma implementação descreve como a estrutura de dados o faz. Uma interface define o conjunto de operações suportado por uma estrutura de dados e a semântica, ou significado, dessas operações. Uma implementação descreve como essas operações são realizadas em termos de código e algoritmos específicos. Compreender as interfaces e implementações ajuda os programadores a escolher a estrutura de dados correta para um problema específico e a escrever código eficiente que usa essa estrutura de dados da maneira mais eficaz possível.

✓ Quais são as vantagens de usar a notação Big-O para analisar algoritmos?

*1/1

- ☐ Permite raciocinar a um nível mais alto
- ☐ Torna possível analisar funções mais complicadas
- ☐ Ajuda a comparar o desempenho de diferentes algoritmos
- ☐ Permite prever o tempo de execução para entradas grandes
- ☒ Todas as anteriores



✓ O que a notação Big-O representa? *

1/1

- ☐ O tempo de execução exato de um algoritmo
- ☐ O número de instruções do computador realizadas durante uma operação
- ☒ Um conjunto de funções que descrevem o comportamento assintótico do tempo de execução
- ☐ A complexidade do código em termos de linhas escritas
- ☐ Nenhuma das anteriores



- ☐ $O(1)$ descreve um algoritmo com tempo de execução constante, enquanto $O(n)$ descreve um algoritmo com tempo de execução linear.
- ☐ $O(1)$ descreve um algoritmo com tempo de execução linear, enquanto $O(n)$ descreve um algoritmo com tempo de execução constante.
- ☐ Ambos descrevem algoritmos com tempo de execução constante.
- ☐ Ambos descrevem algoritmos com tempo de execução linear.
- ☒ Nenhuma das anteriores. ✗

Resposta correta

- ☒ $O(1)$ descreve um algoritmo com tempo de execução constante, enquanto $O(n)$ descreve um algoritmo com tempo de execução linear.

Feedback

A diferença entre $O(1)$ e $O(n)$ é que $O(1)$ descreve um algoritmo com tempo de execução constante, ou seja, o tempo de execução não depende do tamanho da entrada. Já $O(n)$ descreve um algoritmo com tempo de execução linear, ou seja, o tempo de execução cresce proporcionalmente ao tamanho da entrada. Em outras palavras, um algoritmo com complexidade $O(1)$ é considerado mais eficiente do que um algoritmo com complexidade $O(n)$, pois o primeiro tem um tempo de execução constante independentemente do tamanho da entrada, enquanto o segundo tem um tempo de execução que cresce linearmente com o tamanho da entrada.

Este formulário foi criado em FEN UERJ.

Google Formulários

