

01. [Sebesta, 2000] Defina tipo de dados abstrato.
02. [Sebesta, 2000] Quais são as vantagens das duas partes da definição de um tipo de dado abstrato?
03. [Sebesta, 2000] Quais são as exigências de projeto de uma linguagem que suporta tipos de dados abstratos?
04. [Sebesta, 2000] Quais são as questões de projeto de linguagens para tipos de dados abstratos?
05. [Sebesta, 2000] O que falta no suporte da SIMULA 67 para os tipos de dados abstratos?
06. [Sebesta, 2000] Qual recurso das classes SIMULA 67 tornou-se uma parte fundamental das linguagens orientadas a objeto?
07. [Sebesta, 2000] Quais são as duas razões pelas quais os tipos de dados abstratos do Modula-2 restringem-se a ser ponteiros?
08. [Sebesta, 2000] Explique como a ocultação de informação é oferecida em um pacote Ada.
09. [Sebesta, 2000] Qual é a diferença entre tipos `private` e `limited private` na Ada?
10. [Sebesta, 2000] Como os objetos da classe C++ são criados?
11. [Sebesta, 2000] Onde os objetos da classe Java são criados?
12. [Sebesta, 2000] Por que o Java não tem destruidores?
13. [Sebesta, 2000] O que é um construtor? O que é um destrutor?
14. [Sebesta, 2000] O que é uma função-amiga? O que é uma classe amiga?
15. [Sebesta, 2000] Dê uma razão pela qual o Java não tem funções amigas ou classes amigas.
16. [Sebesta, 2000] Como as instâncias das classes modelo C++ são criadas?
17. [Sebesta, 2000] Projete o exemplo de tipo abstrato de pilha em Pascal, supondo que a definição de pilha, as suas operações e o código que a usa estejam todos no mesmo programa.
18. [Sebesta, 2000] Qual parte ou partes críticas da definição de um tipo de dado abstrato estão faltando em uma implementação Pascal do tipo pilha, como por exemplo, a do Problema 17?
19. [Sebesta, 2000] Projete o exemplo de tipo abstrato de pilha em FORTRAN 77, usando um único subprograma com múltiplas entradas para a definição de tipo e as operações.
20. [Sebesta, 2000] Como a implementação FORTRAN do Problema 19 compara-se com a implementação Ada deste capítulo em termos de confiabilidade e de flexibilidade?
21. [Sebesta, 2000] Modifique a classe C++ do tipo abstrato de pilha para usar uma representação de lista encadeada e teste-a com o mesmo código que aparece neste capítulo.
22. [Sebesta, 2000] Alguns engenheiros de software acreditam que todas as entidades importadas devem ser qualificadas pelo nome da unidade de programa exportadora. Você concorda? Sustente sua resposta.
23. [Sebesta, 2000] Projete um tipo de dado abstrato para uma abstração de matriz em uma linguagem que você conheça, incluindo operações de adição, de subtração e de multiplicação de matrizes.
24. [Sebesta, 2000] Projete um tipo de dado abstrato de fila em uma linguagem que você conheça, incluindo operações para enfileirar, desenfileirar e esvaziar.
25. [Sebesta, 2000] Suponhamos que alguém tenha projetado um tipo de dado abstrato de pilha no qual a função, `top`, retorne um caminho (ou ponteiro) de acesso em vez de retornar uma cópia do elemento `top`. Essa não é uma abstração de dados verdadeira. Por quê? Dê um exemplo que ilustre o problema.

26. [Sebesta, 2000] Escreva um tipo de dado abstrato para números complexos, incluindo operações de adição, de subtração, de multiplicação, de divisão, de extração de cada uma das partes de um número complexo e de construção de um número complexo a partir de duas constantes, variáveis ou expressões de ponto-flutuante. Use o Modula-2, a Ada, o C++ ou o Java.
  
27. [Sebesta, 2000] Escreva um tipo de dado abstrato para filas cujos elementos armazenem nomes de 10 caracteres. Os elementos da fila devem ser alocados dinamicamente do *heap*. As operações da fila são enfileirar, desenfileirar e esvaziar. Use o Modula-2, a Ada, o C++ ou o Java.