

Lista de Exercícios — TAD

Tipo Abstrato de Dados · Nível Introdutório · Com Gabarito

Questão 1

O que é um Tipo Abstrato de Dado (TAD)?

- a) Um tipo de dado que só existe em linguagens orientadas a objetos.
- b) Um modelo matemático que define um conjunto de valores e um conjunto de operações sobre esses valores, sem se preocupar com a implementação.
- c) Um tipo primitivo definido pela linguagem de programação, como int e float.
- d) Uma estrutura de dados que organiza informações na memória do computador.

Questão 2

Por que o Tipo Abstrato de Dado é chamado de 'abstrato'?

- a) Porque é difícil de entender e implementar.
- b) Porque sua definição não está vinculada a linguagens de programação nem a implementações específicas.
- c) Porque não pode ser implementado em linguagem C.
- d) Porque seus dados são invisíveis para o compilador.

Questão 3

Qual é a diferença entre um TAD e uma Estrutura de Dados?

- a) Não há diferença, os dois termos significam a mesma coisa.
- b) O TAD é abstrato (define O QUE faz), enquanto a Estrutura de Dados é concreta (define COMO os dados são organizados na memória).
- c) A Estrutura de Dados é mais eficiente que o TAD.
- d) O TAD só pode ser usado em C, enquanto a Estrutura de Dados funciona em qualquer linguagem.

Questão 4

Um TAD é formado por:

- a) Um conjunto de variáveis e um arquivo de cabeçalho.
- b) Uma estrutura de dados e um arquivo .c
- c) Um conjunto de valores (operandos) e um conjunto de operações (operadores) sobre esses valores.
- d) Um conjunto de funções e um conjunto de ponteiros.

Questão 5

Em C, qual mecanismo é normalmente usado para apresentar a interface de um TAD ao usuário?

- a) Arquivos de configuração (.cfg)
- b) Arquivos de cabeçalho (.h) separados do código de implementação (.c)
- c) Funções declaradas diretamente no main
- d) Variáveis globais compartilhadas entre arquivos

Questão 6

Analise o código abaixo e responda: ele representa corretamente um TAD? Justifique.

```
int main() {  
    float x1 = 10, y1 = 20;  
    float x2 = 15, y2 = 30;  
    double d = distancia(x1, y1, x2, y2);  
}
```

- a) Sim, pois há dados e uma operação de distância definida.
- b) Não, pois as variáveis estão soltas e a função de distância pode receber quaisquer quatro números, sem vínculo com um tipo específico.
- c) Sim, pois está implementado em C.
- d) Não, pois falta o arquivo .h correspondente.

Questão 7

Sobre um mesmo TAD, é correto afirmar que:

- a) Só pode ter uma única implementação possível.
- b) Pode ter implementações diferentes, desde que todas respeitem o contrato definido pela interface.
- c) Não pode ser alterado após ser implementado.
- d) Precisa usar ponteiros obrigatoriamente em C.

Questão 8

Observe a interface abaixo do TAD Relogio. Qual das assinaturas de função está INCORRETA para esse TAD?

```
typedef struct relógio Relógio;  
  
Relógio* cria_relogio();  
void verifica_hora(Relógio*, int*, int*, int*);  
void modifica_hora(Relógio*, int, int, int);
```

- a) Relógio* cria_relogio();
- b) void verifica_hora(Relógio*, int*, int*, int*);
- c) int soma_horas(int a, int b);
- d) void modifica_hora(Relógio*, int, int, int);

Questão 9

Complete a implementação da função `cria_relogio()` do TAD `Relogio`, que deve alocar memória e inicializar horas, minutos e segundos com zero:

```
struct relogio{ int H; int M; int S; };
typedef struct relogio Relogio;

Relogio* cria_relogio(){
    // complete aqui
}
```

Questão 10

Qual das afirmações abaixo sobre TADs em C está CORRETA?

- a) Em C é possível tornar os dados do TAD completamente privados, sem nenhuma forma de acesso externo.
- b) Classes em C++ e Java são mais precisas para representar TADs do que cabeçalhos em C, pois permitem definir dados não visíveis ao usuário.
- c) O uso de arquivos `.h` em C garante que o usuário nunca terá acesso aos dados internos do TAD.
- d) TADs só podem ser implementados em linguagens orientadas a objetos.

Gabarito — Lista de TAD

Questao 1

Alternativa correta: B

Um TAD é um modelo matematico que define um conjunto de valores e um conjunto de operacoes sobre eles, sem se preocupar com a implementacao.

Questao 2

Alternativa correta: B

O termo 'abstrato' existe justamente porque a definicao do TAD nao esta vinculada a nenhuma linguagem de programacao ou implementacao especifica.

Questao 3

Alternativa correta: B

O TAD define O QUE o tipo faz (abstrato). A Estrutura de Dados define COMO os dados sao organizados na memoria (concreto). Um TAD se concretiza atraves de uma Estrutura de Dados.

Questao 4

Alternativa correta: C

Um TAD e formado por um conjunto de valores (operandos) e um conjunto de operacoes (operadores) que atuam sobre esses valores.

Questao 5

Alternativa correta: B

Em C, usamos arquivos de cabecalho (.h) para apresentar a interface do TAD — as assinaturas das funcoes — sem expor a implementacao interna.

Questao 6

Alternativa correta: B

Nao e um TAD porque as variaveis estao soltas no codigo e a funcao de distancia pode receber quaisquer quatro numeros, sem vinculo com um tipo especifico.

Questao 7

Alternativa correta: B

Um mesmo TAD pode ter multiplas implementacoes diferentes. O que importa e que todas respeitem o contrato definido pela interface (os dados e as operacoes prometidas).

Questao 8

Alternativa correta: C

A funcao 'int soma_horas(int a, int b)' nao pertence ao TAD Relogio pois nao opera sobre um ponteiro do tipo Relogio, quebrando o encapsulamento.

Questao 9

Resposta esperada:

```
Religio* cria_relogio(){
    Religio *r = (Religio*)malloc(sizeof(Religio));
    r->H = 0;
    r->M = 0;
    r->S = 0;
    return r;
}
```

A funcao deve alocar memoria com malloc, inicializar H, M e S com zero e retornar o ponteiro.

Questao 10

Alternativa correta: B

Classes em C++ e Java sao mais precisas para TADs pois permitem definir dados privados. Em C, o usuario do TAD ainda tem acesso ao conteudo do struct se o .c for exposto.