

Exercícios de fixação 02 - Arquivos de dados estruturados

- Entrega 25 ago em 23:59
- Pontos 1
- Perguntas 4
- Limite de tempo Nenhum

Instruções

Este questionário contém questões sobre os tipos de dados usados em arquivos, sobre codificação de caracteres e sobre registros de tamanho fixo e variável.

Pontuação deste teste: 0,75 de 1

Enviado 21 ago em 14:01

Esta tentativa levou 11 minutos.



Pergunta 1

0 / 0,25 pts

Considere um sistema que armazena, em arquivos, strings de tamanho variável de até 5000 caracteres usando a codificação UTF-8. Cada string, nesse sistema, é armazenada com o seu próprio indicador de tamanho, cujo tamanho deve ser considerado.

Quantos *bytes* serão usados no arquivo por esse sistema para armazenar a string **EDUCAÇÃO**?

Você respondeu

10

12 (com margem: 0)

Um indicador de tamanho, para até 5000 caracteres, deve ter 2 bytes. A string possui 8 caracteres, mas os 2 caracteres acentuados usarão 2 bytes cada um. Assim, serão necessários 12 bytes para armazenar a string.



Pergunta 2

0,25 / 0,25 pts

Por que devemos usar o sistema de codificação UTF-8 no armazenamento de *strings* em arquivos?

Correto!

- ☒ Porque o UTF-8 é um sistema de codificação universal.

- ☐ Porque o UTF-8 é um padrão brasileiro.
- ☐ Porque o UTF-8 usa no máximo 8 bits para representação de símbolos.
- ☐ Porque o UTF-8 é um sistema de codificação de tamanho fixo.

A vantagem do uso do UTF-8 é o fato de ele permitir a representação de qualquer símbolo existente, ou seja, é universal. Além disso, ele é de tamanho variável e acaba consumindo menos bytes que as representações UTF-16 ou UTF-32, também universais.



Pergunta 3

0,25 / 0,25 pts

Qual das seguintes formas é a mais adequada para o armazenamento em arquivo de um atributo data-hora, isto é, de um atributo que armazene um momento específico de uma data específica (ex.: 20/08/2024 19:30). Considere, na sua resposta, que o sistema em que essa data será armazenada deverá ser capaz de indicar quanto tempo já se passou desde esse momento específico.

- ☐ Um atributo *int* (para os dias passados desde uma data inicial) e outro atributo *float* (para a fração de dia que representa as horas e minutos).
- ☐ Uma *string* em que a data e a hora serão armazenados na forma de caracteres.
- ☐ Cinco atributos *int* (para dia, mês, ano, horas e minutos).

Correto!

- ☒ Um atributo *long* (representando os milissegundos passados desde uma data inicial).

A melhor alternativa entre essas é aquela que usa apenas um atributo e que permite operações matemáticas, já que se espera determinar quanto tempo se passou desde a data. Assim, a melhor alternativa é converter todos os valores (datas e horas) para milissegundos passados desde uma data inicial (dia zero). Obviamente, serão necessárias funções de conversão entre esse tipo *long* e um tipo *Date* que esteja disponível na linguagem de programação. Essa conversão seria realizada nas interações com o usuário.



Pergunta 4

0,25 / 0,25 pts

Campos com múltiplos valores são adequados quando o número de valores do atributo é variável. Por exemplo, uma pessoa pode ter 1, 2, 3 ou mais números de telefone, emails, etc. Nesses casos, qual é a forma adequada para se registrar esses valores em um registro de tamanho variável?

- ☐ Usar também o delimitador de campos para separar cada valor desse campo específico.

☐ Reservar uma quantidade de *bytes* no arquivo suficiente para conter vários valores e preencher os *bytes* que sobram com espaços em branco.

☐ Pré-determinar a quantidade máxima de valores e escrever aqueles não usados como espaços em branco no arquivo.

Correto!

- ☒ Escrever um indicador da quantidade de valores no início do campo e, em seguida, cada um dos valores.

A solução correta para campos com múltiplos valores é escrever um único *byte* inicial que contenha a quantidade de valores do campo. Dessa forma, é possível se ler cada um dos valores desse campo, sem confundi-los com os valores de outros campos.

Pontuação do teste: 0,75 de 1