

Exercícios de fixação 02 - Arquivos de dados estruturados

Entrega 18 fev em 9:00**Pontos** 1**Perguntas** 4**Disponível** até 18 fev em 9:00**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.

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	23 minutos	0,75 de 1

❗ As respostas corretas estão ocultas.

Pontuação deste teste: **0,75** de 1

Enviado 16 fev em 15:28

Esta tentativa levou 23 minutos.

Pergunta 1

0,25 / 0,25 pts

Indique quais são os tipos de dados adequados para cada um dos campos de um registro **Cliente**.

Data de nascimento

int

**Número de dependentes**

byte

**CEP**

string

**Renda pessoal**

float



O atributo **Renda pessoal** é um valor monetário, com duas casas decimais. Assim, o tipo adequado para esse atributo é o **float**. O CEP é um campo baseado em dígitos, mas que não é usado em operações aritméticas. Assim, pode ser armazenado como **string**. O atributo número de dependentes conterá sempre um valor pequeno, menor que 128. Portanto, pode ser do tipo **byte**. A **Data de nascimento** pode ser armazenado como um **int**, permitindo que se calcule, com facilidade, a idade atual do cliente.

Pergunta 2

0,25 / 0,25 pts

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



Porque o UTF-8 usa no máximo 8 bits para representação de símbolos.



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



Porque o UTF-8 é um padrão brasileiro.



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.

Incorreta

Pergunta 3**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**?

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 4**0,25 / 0,25 pts**

Por que atributos como CPF e CNPJ devem ser armazenados em arquivos como *strings* ao invés de tipos numéricos?



Porque apenas ao serem armazenados como *strings*, seria possível ordená-los sequencialmente.



Porque esses atributos podem conter caracteres entre os dígitos, como pontos, traços e barras, que seriam perdidos em tipos numéricos.



Porque não há tipos numéricos que comportem todos os dígitos desses atributos e os valores acabariam sendo cortados.



Porque esses atributos não são valores usados em operações matemáticas e os seus dígitos precisam ser separados para verificação.

Valores como CPF e CNPJ, apesar de serem representados apenas com dígitos, não devem ser armazenados como números, porque não são usados em operações matemáticas. Na verdade, seus dígitos devem ser separados para a única operação frequente: a verificação do(s) dígito(s) verificador(es). Para isso, é mais fácil extrair os dígitos de uma *string* do que de um tipo numérico.

Apesar de usarmos vários separadores nesses atributos, eles não precisam ser inseridos nos valores. Eles podem ser usados apenas nas máscaras (de edição ou de apresentação).

Também é importante lembrar que, nesses atributos, os zeros à esquerda são significativos.

Pontuação do teste: **0,75** de 1