

Quiz 02 - Tipos de campos

- Entrega 21 ago em 21:10
- Pontos 6
- Perguntas 6
- Disponível 20 ago em 20:45 - 21 ago em 23:30
- Limite de tempo Nenhum

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	9 minutos	3,5 de 6
Pontuação deste teste: 3,5 de 6			
Enviado 21 ago em 9:33			
Esta tentativa levou 9 minutos.			
⋮			
Pergunta 1			
0,5 / 1 pts			
Indique quais são os tipos de dados adequados para cada um dos campos de um registro Cliente .			
Data de nascimento (sem horas)			
<div>string</div>			
int			
Número de filhos			
<div>int</div>			
byte			
CEP			
<div>string</div>			
Renda pessoal (até 100 mil reais)			
<div>double</div>			

Outras opções de respostas incorretas:

- long
- char

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.

Resposta correta



Pergunta 2

1 / 1 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`?

12

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.

Resposta correta



Pergunta 3

1 / 1 pts

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

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

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.

Resposta incorreta



Pergunta 4

0 / 1 pts

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



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



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



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



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.

Resposta incorreta



Pergunta 5

0 / 1 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.



Uma *string* em que a data e a hora serão armazenados na forma de caracteres.



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).



Cinco atributos *int* (para dia, mês, ano, horas e minutos).



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.

Resposta correta



Pergunta 6

1 / 1 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?



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



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



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.

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: 3,5 de 6