

≡ 02

Avaliando refatorações

Você faz parte do time de desenvolvimento de uma aplicação de vendas de carro.

Observe o bloco de código abaixo, em que são obtidos o volume de vendas, o preço máximo e o preço mínimo de negociações dessa aplicação:

```
foreach (Negociacao negociacao in negociacoes)
{
    volume += negociacao.GetVolume();

    if (negociacao.GetPreco() > maximo) {
        maximo = negociacao.GetPreco();
    }
    if (negociacao.GetPreco() < minimo) {
        minimo = negociacao.GetPreco();
    }
}
```

Como você utilizaria refatoração no código acima?

Escolha a melhor resposta.

Selezione uma alternativa

A

```
foreach (Negociacao negociacao in negociacoes)
{
    var volumeVendas = negociacao.GetVolume();

    volume += volumeVendas;

    if (negociacao.GetPreco() > maximo) {
        maximo = negociacao.GetPreco();
    }
    if (negociacao.GetPreco() < minimo) {
        minimo = negociacao.GetPreco();
    }
}
```

B

```
foreach (Negociacao negociacao in negociacoes)
{
    var volumeVendas = negociacao.GetVolume();

    volume += volumeVendas;

    var preco = negociacao.GetPreco();
    if (preco > maximo) {
        maximo = preco;
    }
    if (preco < minimo) {
        minimo = preco;
    }
}
```

```
    }  
}
```

C

```
foreach (Negociacao negociacao in negociacoes)  
{  
    volume += negociacao.GetVolume();  
  
    var preco = negociacao.GetPreco();  
    if (preco > maximo) {  
        maximo = preco;  
    }  
    if (preco < minimo) {  
        minimo = preco;  
    }  
}
```

D

```
foreach (Negociacao negociacao in negociacoes)  
{  
    volume += negociacao.GetVolume();  
  
    var preco = negociacao.GetPreco();  
  
    maximo = GetMaximo(preco, maximo);  
    minimo = GetMinimo(preco, minimo);  
}  
...  
  
decimal GetMaximo(decimal preco, decimal maximo)  
{  
    if (preco > maximo) {  
        return preco;  
    }  
    else  
    {  
        return maximo;  
    }  
}  
  
decimal GetMinimo(decimal preco, decimal minimo)  
{  
    if (preco < minimo) {  
        return preco;  
    }  
    else  
    {  
        return minimo;  
    }  
}
```