

Code Challenge: Ganho de Capital

Contexto

O objetivo deste exercício é implementar um programa que calcula o imposto a ser pago sobre lucros ou prejuízos de operações no mercado financeiro de ações.

Por favor, leia as instruções abaixo e sinta-se à vontade para fazer perguntas, caso ache necessário.

Exemplo de uso do Ganho de Capital

Como o programa deve funcionar?

Entrada

Seu programa vai receber uma lista de operações do mercado financeiro de ações em formato `json` através da entrada padrão (`stdin`). Cada operação desta lista contém os seguintes campos:

Nome	Significado
<code>operation</code>	Se a operação é uma operação de compra (<code>buy</code>) ou venda (<code>sell</code>)
<code>unit-cost</code>	Preço unitário da ação
<code>quantity</code>	Quantidade de ações negociadas

Este é um exemplo da entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 10000}, {"operation": "sell", "unit-cost": 20, "quantity": 5000}]
```

A última linha da entrada será uma linha vazia.

Saída

O programa deve retornar uma lista contendo o imposto pago para cada operação recebida. Os elementos desta lista devem estar codificados em formato `json` e a saída deve ser retornada através da saída padrão (`stdout`). O retorno é composto pelo seguinte campo:

Nome	Significado
<code>tax</code>	O valor do imposto pago em uma operação

Este é um exemplo da saída:

```
[{"tax": 0}, {"tax": 10000}]
```

A lista retornada pelo programa deve ter o mesmo tamanho da lista de operações processadas na entrada. Por exemplo, se foram processadas três operações (buy, buy, sell), o retorno do programa deve ser uma lista com três valores que representam o imposto pago em cada operação.

Regras do Ganho de Capital

O programa deve lidar com dois tipos de operações (`buy` e `sell`) e ele deve seguir as seguintes regras:

- Nenhum imposto é pago em operações de compra;
- Para determinar se a operação resultou em lucro ou prejuízo, você pode calcular o **preço médio ponderado**, onde o peso corresponde ao número de ações compradas com determinado preço. Por exemplo, se você comprou 10 ações por R\$ 20 e 5 ações por R\$ 10, o preço médio ponderado é $(10 \times 20 + 5 \times 10) / 15 = 16.66$.
- Prejuízos acontecem quando você vende ações a um valor menor do que o preço médio ponderado de compra. Neste caso, nenhum imposto deve ser pago e você deve subtrair o prejuízo dos lucros seguintes, antes de calcular o imposto.
- Você não paga nenhum imposto se o valor total da operação (custo unitário da ação x quantidade) for menor ou igual a R\$ 20000. Use o valor total da operação e não o lucro obtido para determinar se o imposto deve ou não ser pago. E não se esqueça de deduzir o prejuízo dos lucros seguintes.
- O percentual de imposto pago é de 20% sobre o lucro obtido. Ou seja, o imposto vai ser pago quando há uma operação de venda cujo preço é superior ao preço médio ponderado de compra.
- Você deve usar o prejuízo passado para deduzir múltiplos lucros futuros, até que todo prejuízo seja deduzido.

Você pode assumir que nenhuma operação vai vender mais ações do que você tem naquele momento.

Exemplos do Ganho de Capital

Caso #1

Operação	Custo unitário	Quantidade	Imposto Pago	Explicação
<code>buy</code>	10	100	0	Comprar ações não paga imposto
<code>sell</code>	15	50	0	Valor total menor do que R\$ 20000
<code>sell</code>	15	50	0	Valor total menor do que R\$ 20000

Entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 100}, {"operation": "sell", "unit-cost": 15, "quantity": 50}, {"operation": "sell", "unit-cost": 15, "quantity": 50}]
```

Saída:

```
[{"tax": 0}, {"tax": 0}, {"tax": 0}]
```

Caso #2

Operação	Custo unitário	Quantidade	Imposto Pago	Explicação
buy	10	10000	0	Comprar ações não paga imposto
sell	20	5000	10000	Lucro de R\$ 50000: 20% do lucro corresponde a R\$ 10000 e não possui prejuízo anterior
sell	5	5000	0	Prejuízo de R\$ 25000: não paga imposto

Entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 10000}, {"operation": "sell", "unit-cost": 20, "quantity": 5000}, {"operation": "sell", "unit-cost": 5, "quantity": 5000}]
```

Saída:

```
[{"tax": 0}, {"tax": 10000}, {"tax": 0}]
```

Caso #3

Operação	Custo unitário	Quantidade	Imposto Pago	Explicação
buy	10	10000	0	Comprar ações não paga imposto
sell	5	5000	0	Prejuízo de R\$ 25000: não paga imposto
sell	20	5000	5000	Lucro de R\$ 50000. Deve deduzir prejuízo de R\$ 25000: 20% de R\$ 25000 -> R\$ 5000

Entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 10000}, {"operation": "sell", "unit-cost": 5, "quantity": 5000}, {"operation": "sell", "unit-cost": 20, "quantity": 5000}]
```

Saída:

```
[{"tax": 0}, {"tax": 0}, {"tax": 5000}]
```

Caso #4

Operação	Custo unitário	Quantidade	Imposto Pago	Explicação
buy	10	10000	0	Comprar ações não paga imposto
buy	25	5000	0	Comprar ações não paga imposto
sell	15	10000	0	Considerando preço médio ponderado de R\$ 15 -> não teve lucro nem prejuízo

Entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 10000}, {"operation": "buy", "unit-cost": 25, "quantity": 5000}, {"operation": "sell", "unit-cost": 15, "quantity": 10000}]
```

Saída:

```
[{"tax": 0}, {"tax": 0}, {"tax": 0}]
```

Caso #5

Operação	Custo unitário	Quantidade	Imposto Pago	Explicação
buy	10	10000	0	Comprar ações não paga imposto
buy	25	5000	0	Comprar ações não paga imposto
sell	15	10000	0	Considerando preço médio ponderado de R\$ 15 -> não teve lucro nem prejuízo
sell	25	5000	10000	Considerando preço médio ponderado de R\$ 15 -> lucro de R\$ 50000; 20% do lucro é R\$ 10000

Entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 10000}, {"operation": "buy", "unit-cost": 25, "quantity": 5000}, {"operation": "sell", "unit-cost": 15, "quantity": 10000}, {"operation": "sell", "unit-cost": 25, "quantity": 5000}]
```

Saída:

```
[{"tax": 0}, {"tax": 0}, {"tax": 0}, {"tax": 10000}]
```

Caso #6

Operação	Custo unitário	Quantidade	Imposto Pago	Explicação
buy	10	10000	0	Comprar ações não paga imposto
sell	2	5000	0	Perda de R\$ 40000: valor total é menor do que R\$ 20000. Mas devemos deduzir o prejuízo independente disso.
sell	20	2000	0	Lucro de R\$ 20000: se deduzir o prejuízo, seu lucro é R\$ 0 e você ainda tem R\$ 20000 de prejuízo para deduzir dos próximos lucros.
sell	20	2000	0	Lucro de R\$ 20000: se deduzir o prejuízo, seu lucro é R\$ 0 e não tem mais prejuízo para deduzir dos próximos lucros.
sell	25	1000	3000	Lucro de R\$ 15000 e sem prejuízos: imposto de 20% sobre o lucro.

Entrada:

```
[{"operation": "buy", "unit-cost": 10, "quantity": 10000}, {"operation": "sell", "unit-cost": 2, "quantity": 5000}, {"operation": "sell", "unit-cost": 20, "quantity": 2000}, {"operation": "sell", "unit-cost": 20, "quantity": 2000}, {"operation": "sell", "unit-cost": 25, "quantity": 1000}]
```

Saída:

```
[{"tax": 0}, {"tax": 0}, {"tax": 0}, {"tax": 0}, {"tax": 3000}]
```

Estado da aplicação

O programa **não deve depender** de nenhum banco de dados externo e o estado interno da aplicação deve ser gerenciado em memória explicitamente por alguma estrutura que achar adequada. O estado da aplicação deve estar vazio sempre que a aplicação for inicializada.

Lidando com erros

Você pode assumir que não ocorrerão erros na conversão do `json` de entrada. Na avaliação da sua solução nós não vamos utilizar entradas que contenham erros, estejam mal formatadas, ou que quebrem o contrato.

Nossas Expectativas

Nós no Nubank valorizamos as seguintes qualidades:

- **Simplicidade:** espera-se da solução um projeto pequeno e de fácil entendimento;

- **Elegância:** espera-se da solução facilidade de manutenção, uma separação clara das responsabilidades e uma estrutura de código bem organizada;
- **Operacional:** espera-se da solução a resolução do problema, seus casos de borda ou extremos e a capacidade de extensão para futuras decisões de design.

Desta forma, procuraremos avaliar:

- Uso adequado de [trasparência referencial](#) quando aplicável;
- Testes de unidade e integração de qualidade;
- Documentação onde for necessário;
- Instruções sobre como executar o código.

Por último, porém não menos importante:

- Você pode utilizar bibliotecas de código aberto (open source) que acredite serem adequadas para ajudar na solução do desafio, por exemplo analisadores de json; Por favor tente limitar o uso de frameworks e [boilerplate code](#) desnecessários.
- O desafio espera uma aplicação de linhas de comando **independente**; Por favor, evite adicionar infraestrutura desnecessária e/ou dependências externas. É esperado que você seja capaz de identificar as ferramentas necessárias para resolver o problema apresentado sem adicionar camadas extras de complexidade.


Notas gerais

- Esse desafio poderá ser estendido por você e por outra pessoa engenheira do Nubank durante uma outra etapa do processo;
- O Ganho de Capital deve receber as operações através da entrada padrão (`stdin`) e retornar o resultado do processamento através da saída padrão (`stdout`), ao invés de uma API REST.

Preparando seu desafio para envio

- Você deve entregar o código fonte de sua solução para nós em um arquivo comprimido (zip) contendo o código e toda documentação possível. Favor não incluir arquivos desnecessários como binários compilados, bibliotecas, etc;
- Não faça o upload da sua solução em nenhum repositório público como GitHub, BitBucket, etc;

Remova informações pessoais

 **IMPORTANTE:** Por favor remova toda informação que possa lhe identificar nos arquivos do desafio antes de enviar a solução. Atenção especial para os seguintes pontos:

- Arquivos da solução como código, testes, namespaces, binários, comentários, e nomes dos arquivos;
- Comentários automáticos que seu editor de código pode ter adicionado aos arquivos;
- Documentação do código como annotations, metadata, e README.MD;
- Informações de autoria do código e configuração do versionador de código.

Se você planeja utilizar [git](#) como sistema de controle de versões, execute o seguinte comando na raíz do repositório para exportar a solução anonimizada:

```
git archive --format=zip --output=./capital-gains.zip HEAD
```

Adicione um README

Sua solução deve conter um arquivo de README com:

- Uma explicação sobre as decisões técnicas e arquiteturais do seu desafio;
- Uma justificativa para o uso de frameworks ou bibliotecas (caso sejam usadas);
- Instruções sobre como compilar e executar o projeto;
- Notas adicionais que você considere importantes para a avaliação.

Ambiente de execução

O processo de build e execução da aplicação deve ser possível num sistema operacional Unix ou Mac. [Builds containerizadas](#) são bem vindas.