

Desafio técnico - 1/2022

O que queremos medir?

- Inglês
- Fundamentos de programação
 - Domínio de uma linguagem de programação
 - Estrutura de dados
 - Tipos de dados (int, float, string)
 - Tipos encadeados de dados (listas, tuplas, conjuntos, hashmaps)
 - Orientação a objetos
 - Classe
 - Método
 - Atributo
 - Construtores
 - Programação funcional
 - Imutabilidade
 - Recursão
 - Sistema de Tipos
- Capacidade de buscar por soluções

Desafios

Estes são os desafios que você pode tentar resolver. Nós não esperamos que você consiga fazer todos (mesmo um desenvolvedor Sr. teria dificuldades de resolver tudo em um mero fim de semana), portanto resolva quantos desafios você quiser e conseguir.

Algumas informações importantes:

1. As respostas de cada um dos desafios devem estar em pastas distintas, denominadas `desafio_1`, `desafio_2` e `desafio_3`.
2. Não se identifique em nenhum dos arquivos. **Caso qualquer arquivo contenha informações que possam lhe identificar você será eliminado(a) automaticamente.**
3. O código de todos os desafios respondidos deverá ser enviado em um arquivo ZIP nomeado `respostas_desafios.zip` e enviado para o e-mail sol@pandhora.com
4. Os critérios de correção para todos os desafios serão:
 - a. Organização, documentação e clareza do código
 - b. Corretura da solução
5. Em cada desafio a sua pontuação final poderá ser multiplicada por um fator caso você utilize:
 - a. Linguagem de implementação
 - i. Português: 1x
 - ii. Inglês: 1.1x
 - b. Linguagem de programação
 - Python, Java, Go, JavaScript: 1.1x
 - Elixir, Erlang, Scala, C#, TypeScript, Swift: 1.2x
 - C, C++: 1.3x
 - Rust, Clojure ou LISP, Groovy: 1.4x
 - Haskell, F#, OCaml, Idris: 1.6x
 - Outras: 1x
 - c. Input
 - Input via csv: 1.0x
 - Input via XLS: 1.2x
 - Input de funções "hard coded": 1x
 - Input de funções via CLI: 1.1x
 - Inputs via API Rest: 1.5x
 - d. Output
 - Output no terminal: 1x
 - Output em um csv: 1.1x
 - Output em XLS: 1.2x
 - Output via API Rest: 1.5x

Desafio 1 [100 pontos]: Pequenos desafios

Neste desafio queremos medir o seu domínio da linguagem de programação escolhida e de estruturas de dados básicas.

Mini-desafio 1 [25 pontos]: Implemente uma função que dada uma entrada n esta calcule o n -ésimo número da sequência de Fibonacci

Mini-desafio 2 [25 pontos]: Implemente uma função que encontre o MDC (Máximo Divisor Comum) entre dois números inteiros positivos x e y

Mini-desafio 3 [25 pontos]: Implemente uma função que determina se um dado número inteiro positivo x é primo

Mini-desafio 4 [25 pontos]: Implemente uma função que determina se uma dada string é um palíndromo

Desafio 2 [200 pontos] : Sistema de cálculo de retorno acumulado, com condições especiais

Neste desafio vamos medir a sua capacidade de interpretar e implementar requisitos e aumentar progressivamente a complexidade do código.

Junto a este desafio você receberá os arquivos `prices.csv` e `prices.xml`. Caso não tenha recebido os solicite para Luiz Sol em sol@pandho.com.

Etapa 1 [50 pontos]: Calcular retornos diários

Desenvolva um sistema que receba como *input* os dados históricos do valor de uma ação e produza como *output* o uma série histórica dos retornos diários para aquele ativo.

O retorno diário de uma ação em um dia d é igual a $P_d / P_{d-1} - 1$, onde P_d é o preço da ação no dia d e P_{d-1} é o preço da ação no dia anterior ($d-1$).

Exemplo:

Data	Preço	Retorno Diário
2022-01-01	10	Nenhum (não há retorno diário, pois não há preço para o dia anterior)
2022-01-02	12	0.2
2022-01-04	11	-0.0833
2022-01-05	6	0.8333
2022-01-06	24	3

O sistema deverá ler o arquivo `prices.csv` (ou `prices.xml` para mais pontos) que conterà a série histórica de preços e deverá apresentar os dados computados no próprio terminal ou salvá-los no arquivo `daily_returns.csv` (ou `daily_returns.xml` para mais pontos).

Etapa 2 [50 pontos]: Calcular retorno acumulado entre duas datas

Adicione a capacidade do sistema calcular o retorno acumulado entre duas datas quaisquer. Caso a data de início ou de fim da janela selecionada não exista alerte o usuário com um erro ou escolha a data futura mais próxima (+ 5 pontos), caso ela exista.

O retorno acumulado entre duas datas se calcula exatamente como o retorno diário, porém substituindo-se o valor de P_{d-1} por aquele da data de início e o valor de P_d por aquele da data de fim.

No exemplo da tabela anterior, o retorno acumulado entre 2022-01-02 e 2022-01-05 é igual a $6/12 - 1 = -0.5$.

O valor da data de início de começo e fim poderá ser obtida de duas formas

- Variáveis "*hard-coded*": crie duas variáveis globais, uma com nome `window_start_date` e outra com o nome `window_end_date` e leia delas o valor de início e fim da janela
- Parâmetro de execução: nessa forma o programa deverá receber como parâmetro de execução os valores da janelas (ex: `python programa.py 2022-01-10 2022-01-20` ou `programa.exe 2022-01-10 2022-01-20`)

Nos dois casos o retorno acumulado entre as duas datas deve ser impresso no terminal e esta funcionalidade deve existir junto com a da etapa 1, ou seja, o programa deve continuar calculando os retornos diários.

Etapa 3 [100 pontos]: Calcular retorno acumulado janelado

Nesta etapa, juntamente com os retornos diários, para cada dia calcule também o retorno acumulado em uma semana e o acumulado **desde o começo da semana**.

Para calcular o retorno acumulado em uma semana em uma data x você deve determinar uma data de início y onde $y = x - (7 \text{ dias})$, e caso esta data não exista na série histórica utilize a data futura a esta mais próxima, então calcule o retorno acumulado entre y e x

Já para calcular o retorno acumulado desde o começo da semana em uma data x você deve determinar uma data de início y onde y é o último dia útil da semana imediatamente anterior à semana da data x , então calcule o retorno acumulado entre y e x .

Exemplo:

Data	Valor	Retorno diário	Retorno Semanal	Retorno na Semana
2022-02-07 (segunda-feira)	10	Não há	Não há	Não há

2022-02-08 (terça-feira)	11	11 / 10 - 1	Não há	Não há
2022-02-09 (quarta-feira)	12	12 / 11 - 1	Não há	Não há
2022-02-10 (quinta-feira)	13	13 / 12 - 1	Não há	Não há
2022-02-14 (segunda-feira)	14	14 / 13 - 1	14 / 10 - 1	14 / 13 - 1
2022-02-15 (terça-feira)	15	15 / 14 - 1	15 / 11 - 1	15 / 13 - 1
2022-02-16 (quarta-feira)	16	16 / 15 - 1	16 / 12 - 1	16 / 13 - 1
2022-02-17 (quinta-feira)	17	17 / 16 - 1	17 / 13 - 1	17 / 13 - 1
2022-02-18 (sexta-feira)	18	18 / 17 - 1	18 / 14 - 1	17 / 13 - 1
2022-02-21 (segunda-feira)	19	19 / 18 - 1	19 / 14 - 1	19 / 18 - 1

Desafio 3 [400 pontos]: Mini Exchange

Neste desafio nós julgaremos a sua capacidade de estruturar e implementar um sistema a partir da sua descrição funcional, bem como sua capacidade de pesquisa e criatividade.

Instruções

Você deverá implementar um sistema que simule um *Limit Order Book*. Você tem total liberdade para decidir como serão os inputs e *outputs* do sistema.

Aqui (<https://www.youtube.com/watch?v=laiw5iGjXbw>) está um tutorial explicando como funciona esse tipo de sistema.

Se limite a implementar suporte a:

- Adição de ordens limite
- Adição de ordens a mercado
- Cancelamento de ordens