

Introdução

Na aula prática de hoje, vamos considerar o problema da quitanda para ilustrar, além do uso da estrutura de dados dicionário para representar o estoque da quitanda, também como processar arquivos de saída. Suponha que o dicionário estoque dado abaixo esteja povoado com os seguintes dados:

```
estoque = { 'tomate': [1000, 2.30],  
            'alface': [500, 0.45],  
            'batata': [2150, 1.20],  
            'feijão': [100, 5.50] }
```

A chave no dicionário corresponde ao nome do produto e o respectivo valor é uma lista contendo a quantidade do produto no estoque e, em seguida, o preço por unidade do produto.

Escreva um programa para atualizar o estoque a partir de uma lista de transações contendo os dados das vendas do dia, digamos. Uma lista de vendas pode ser:

```
vendas = [ ('tomate', 15.250), ('batata', 10.5), ('alface', 5) ]
```

Cada elemento da lista é uma tupla contendo o produto vendido e a quantidade vendida dele.

Depois o programa deve calcular o valor total das vendas em reais, produzir um relatório das vendas e, finalmente, salvar em novo arquivo o estoque atualizado, utilizando o formato csv (*comma-separated values*). Siga as instruções abaixo para elaborar o programa.

Instruções

1. Faça o download do arquivo p07.py e abra o IDLE com ele. Não se esqueça de salvá-lo de tempos em tempos, porque pode ocorrer falha de energia elétrica durante a aula prática.
2. Digite os comentários obrigatórios (nome, matrícula, data e uma breve descrição sobre o que o programa faz).
3. Estruture seu programa em cinco funções: `main()`, `leiaEstoque(nomearq)`, `leiaVendas(nomearq)`, `processeVendas(estoque, vendas)` e `salveEstoqueAtualizado(nomearq, estoque)`.
4. A função `main` deve ler os dados do estoque usando a função `leiaEstoque(nomearq)`. Depois, deve ler os dados das vendas por meio da função `leiaVendas(nomearq)`. Após, deve chamar a função `processeVendas(estoque, vendas)` para atualizar o estoque de acordo com as vendas realizadas e imprimir um relatório das vendas em forma de tabela. Finalmente, deve interagir com o usuário perguntando-lhe qual é o nome do arquivo de saída que conterá o estoque atualizado e, em seguida, deve chamar a função `salveEstoqueAtualizado(nomearq, estoque)`. Para a formatação da saída, veja o resultado da execução do programa abaixo. Observe que os produtos devem ser salvos em ordem alfabética (crescente) das chaves. A função `main` está pronta; favor não mexer nela.
5. A função `leiaEstoque(nomearq)` recebe o nome do arquivo que contém os dados do estoque, gera o dicionário contendo os dados conforme a estrutura dada na Introdução e, enfim, retorna esse dicionário.

Prática 07 – INF101 – 2023/2 – 2,5 pontos

6. A função `leiaVendas(nomearq)` lê os dados das vendas a partir do arquivo cujo nome é passado como parâmetro, gera a lista de vendas conforme a estrutura dada acima na Introdução e, finalmente, retorna essa lista.
7. A função `processeVendas(estoque, vendas)` recebe o dicionário `estoque` e a lista `vendas` como parâmetros. Depois, faz a atualização do `estoque` de acordo com as transações na lista de vendas e não retorna nada, pois as atualizações são refletidas no próprio parâmetro `estoque` que é passado, naturalmente, por referência. Além disso, esta função deve produzir um relatório das vendas na forma de tabela. Para a formatação da tabela e saber o que deve ser exibido no relatório, inspecione o Exemplo de Execução do Programa abaixo. Caso haja, na lista de vendas, um produto inexistente no `estoque`, a função deve informar que o produto não existe e ignorá-lo para o cálculo do valor total das vendas. Observe a saída da execução do programa abaixo.
8. A função `salveEstoqueAtualizado(nomearq, estoque)` recebe o nome do arquivo de saída que conterá o `estoque` atualizado e o dicionário que contém o `estoque` já atualizado. O arquivo deverá ser escrito por meio de linhas em ordem alfabética (crescente) dos nomes dos produtos. Cada linha tem o nome do produto, a quantidade desse produto no `estoque` e o preço por unidade do respectivo produto. Os dados devem ser separados por vírgula, ou seja, o formato do arquivo deve ser `csv`.
9. Não se esqueça de fechar os arquivos após a utilização deles. Esta ação é importantíssima principalmente para o arquivo de saída, pois assim garante-se que o último *buffer* do arquivo seja descarregado para o disco onde se encontra o arquivo fisicamente.
10. Não se esqueça de chamar a função `main` no final de seu código fonte para iniciar todo o processo.
11. Para testar sua implementação, baixe os arquivos de dados: `estoque.txt` e `vendas.txt` que estão disponíveis no *site* de entrega de trabalhos do LBI. Esses arquivos usam a formatação `csv`. A ordem dos dados nas linhas do arquivo `estoque.txt` é: nome do produto, quantidade em unidades do produto e preço de uma unidade do produto. A ordem dos dados no arquivo `vendas.txt` é: nome do produto vendido e quantidade vendida.
12. Na aula de hoje, para acelerar o processo de depuração do programa, não se exigirá que a função `main` interaja com o usuário para pedir-lhe o nome dos arquivos de entrada. Use diretamente os nomes dados no passo 11 na hora de chamar as respectivas funções que necessitem deles. Porém a interação para pedir o nome do arquivo de saída é preciso.

Após certificar-se de que seu programa esteja correto, envie o arquivo do programa fonte (`p07.py`) através do sistema de entrega do LBI.

Exemplo de Execução do Programa

12 linhas lidas do arquivo 'estoque.txt'.

7 linhas lidas do arquivo 'vendas.txt'.

*****Vendas Realizadas*****

Produto	Quantidade	Preço por Unid.	Valor da Venda
tomate	15.250	2.30	35.07
batata	10.500	1.20	12.60
alface	5.000	0.45	2.25
champignon	0.200	57.80	11.56
chuchu	***não há no estoque***		-----
fubá	15.250	0.65	9.91
feijão	5.500	5.50	30.25
TOTAL			101.64

Estoque atualizado em 6 produtos vendidos.

Entre com o nome do arquivo para conter o estoque atualizado: **estoque_atual.txt**
 Arquivo atualizado e gravado com sucesso contendo 12 linhas.

Conteúdo do Arquivo de Saída: estoque_atual.txt

```
alface,495.000,0.45
batata,2139.500,1.20
champignon,9.800,57.80
couve,250.000,0.45
feijão,94.500,5.50
fubá,934.750,0.65
laranja,250.000,1.54
maçã,100.000,12.35
salsa,100.000,0.30
tomate,984.750,2.30
tomilho,10.000,12.35
uva,50.000,10.37
```