

Introdução

Vimos, em aula teórica, que uma forma de representar os dados de um funcionário de uma empresa é por meio de uma tupla. Por exemplo, considere uma empresa em que cada funcionário seja registrado com os seguintes dados:

- número de matrícula
- nome
- cargo
- salário por hora

Dessa forma, a 4-tupla (2345, 'Ana Lima', 'diretora', 45.72) representa a funcionária Ana Lima cujo número de matrícula é 2345, ocupa o cargo de diretora e seu salário por hora é de R\$45,72. Para representar o banco de dados contendo todos os funcionários da empresa, podemos usar uma lista dessas tuplas. Uma lista é interessante para esse objetivo, pois, se a empresa contratar um novo funcionário, bastará acrescentar sua respectiva tupla no final da lista e, se a empresa dispensar um funcionário, bastará apagar sua respectiva tupla da lista. Para povoar o banco de dados com os dados dos funcionários, realizaremos a leitura dos dados usando arquivos de entrada previamente digitados por meio de algum editor de texto puro. Vamos elaborar o programa de modo a gerar uma listagem dos salários brutos mensais de todos os funcionários da empresa, dado o número de horas trabalhadas no mês de cada funcionário. Além disso, para o cálculo do salário bruto, vamos considerar, além do número de horas regulares trabalhadas no mês, também o número de horas extras no mês. Esta informação é digitada juntamente com o número de horas regulares trabalhadas em uma linha para cada funcionário. Para alcançar este objetivo, siga as instruções abaixo.

Instruções

1. Faça o *download* do esqueleto do programa que se encontra no sistema de entrega do LBI. Abra o IDLE e abra o arquivo fonte `p06.py` onde se encontra o esqueleto. Não se esqueça de salvá-lo de tempos em tempos, porque pode ocorrer falha de energia elétrica durante a aula prática.
2. Complete os comentários obrigatórios (nome, matrícula, data e uma breve descrição sobre o que o programa faz).
3. O programa está estruturado em três funções: `main()`, `leiaFunc(nomeArq)` e `calcSalBruto(nomeArq, bd)`.
4. A função `main()` faz o seguinte: lê os dados de todos os funcionários por meio da função `leiaFunc` que retorna o banco de dados dos funcionários como uma lista de tuplas (veja o passo 5 abaixo); em seguida, na mesma ordem em que os funcionários lidos na primeira função, calcula os salários brutos de todos os funcionários por meio da função `calcSalBruto` (veja o passo 6 a seguir) que os armazena em uma lista e retorna essa lista; e finalmente imprime a tabela dos salários brutos. A função `main` já está pronta no esqueleto. Favor não a modificar.
5. A função `leiaFunc(nomeArq)` tem um parâmetro do tipo *string* para conter o nome externo do arquivo com os dados de todos os funcionários. Cada linha do arquivo de entrada contém o número de matrícula, o nome, o cargo e o salário por hora de um funcionário. Os dados são separados por vírgula. Retire o caractere nova linha que termina cada linha antes de processar os dados contidos na linha. Depois de extrair os dados de uma linha, adicione os quatro dados

Prática 06 – INF101 – 2023/2 – 2,5 pontos

como uma 4-tupla em uma lista. A função retorna esta lista de 4-tuplas com os dados lidos dos funcionários.

6. Implemente a função `calcSalBruto(nomeArq, bd)` com dois parâmetros: o parâmetro `nomeArq` é um *string* para conter o nome externo do arquivo contendo os números de horas trabalhadas regulares e os números de horas extras no mês; o parâmetro `bd` contém o banco de dados de todos os funcionários na forma de uma lista de 4-tuplas. Cada linha do arquivo de entrada contém as horas trabalhadas regulares e as horas extras de um funcionário. Numa linha do arquivo, os dados são separados por um espaço em branco. Para a leitura, inspire-se no que foi feito na função `leiaFunc` acima: leia uma linha de cada vez retirando o caractere ‘\n’ do fim de cada linha e, depois, faça o *split* dos dados agora baseado no caractere espaço em branco, ‘ ’. Para calcular o salário bruto, considere que cada hora extra vale 1,5 do salário por hora do funcionário. A função retorna a lista completa com todos os salários brutos na mesma ordem dos funcionários no banco de dados `bd`.
7. Teste seu programa com os arquivos de entrada fornecidos no *site* de entrega do exercício: `funcionarios.csv` e `horas_trab.dat`. Veja o conteúdo desses arquivos para observar as formatações deles.
8. Se seu programa entrar em *laço infinito* ou travar por alguma razão, digite CTRL-C na janela do *Python Shell* para interromper a execução do programa.

👉 Não se esqueça de preencher o cabeçalho do código fonte com seus dados, a data de hoje e uma breve descrição do programa.

Após certificar-se de que seu programa esteja correto, envie o arquivo com o código fonte (`p06.py`) através do sistema de entrega do LBI.

Exemplo de teste do programa

```
***      Relatório dos Salários Brutos      ***
Matrícula      Nome      Salário Bruto
2345      Ana Lima      11430.00
9876      Pedro Coura      1018.11
3421      Ivo Silva      9489.30
5390      Maria Corrêa      1208.77
7492      João Pedro Gouvêa      3857.28
6537      Alaor Venturini      8515.20
1570      Belarmino Pimenta      14497.60
4906      Ézio Rossi      9278.48
8650      Igor Nunes      9900.00
4217      Natália Alves      9592.10
```