Trabalho prático 1 - Curso de Python
Prof. Mário Carvalho

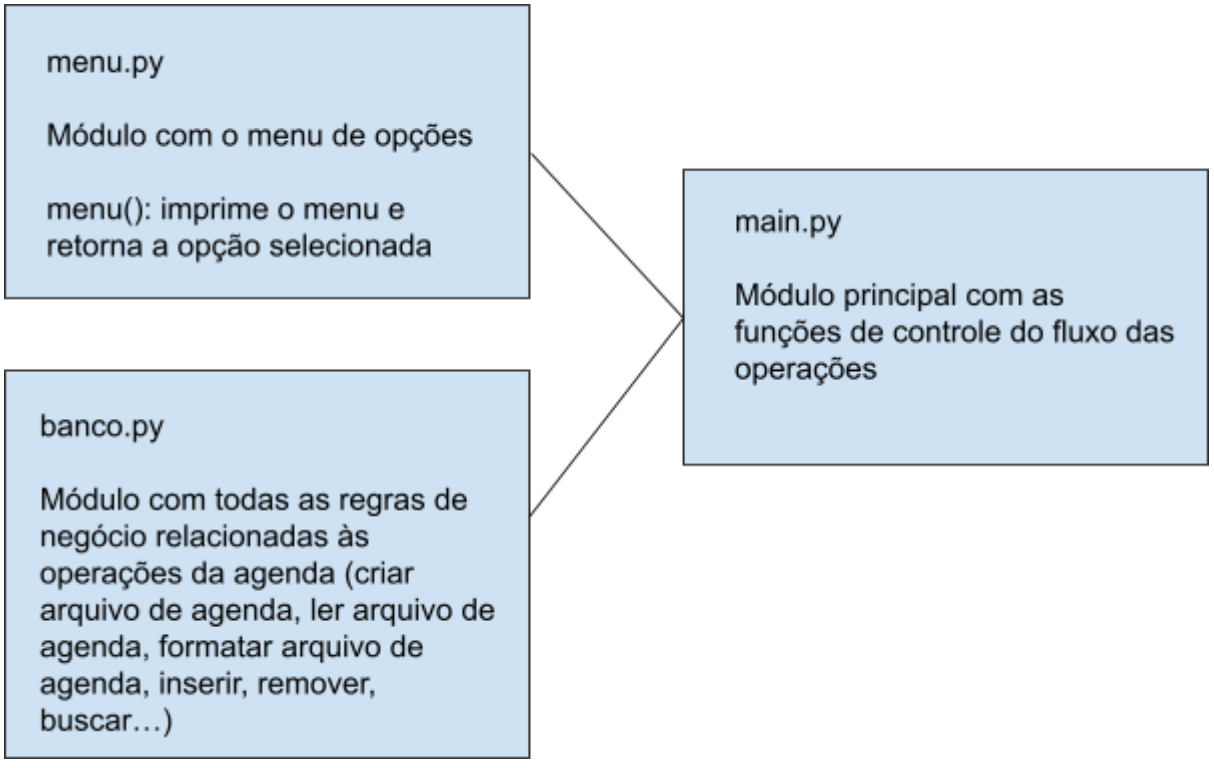
Versão do documento: 3.0
Repositório do professor:
https://github.com/MarioCarvalhoBr/curso-python-2022

Agenda de Contatos

Objetivo:
O objetivo deste trabalho é exercitar todos os assuntos e conceitos vistos até agora durante o curso.

Descrição:
O presente trabalho consiste em criar uma agenda de contatos completa, que simule operações realizadas em agendas de contatos de maneira mais fiel possível ao que temos em nossos smartphones e computadores.

Diagrama com a descrição dos módulos do programa:



Requisitos do programa:

1. **Módulo menu:**
    a. O módulo menu.py consiste em um módulo para retornar o menu de opções a ser exibido para o usuário.

b. Crie uma função chamada menu que imprima o menu de opções. Após mostrar o menu de opções, leia o usuário e retorne a opção selecionada.
c. As opções do menu são:

```
(1) Inserir contato:
(2) Atualizar contato:
(3) Mostrar agenda:
(4) Buscar contato(cpf):
(5) Buscar contato(email):
(6) Buscar contato(nome):
(7) Buscar contato(curso):
(8) Quantidade de contatos:
(9) Deletar contato (cpf):
(10) Deletar contato (email):
(11) Salvar e sair
```

2. **Módulo banco:**
   a. Deve conter todas as regras de negócio relacionadas a agenda, desde a criação do arquivo até as regras para salvar, deletar e listagem.
   b. Deve criar um arquivo chamado **agenda.csv** caso não exista. DICA: Use a opção de arquivo **a+**.
   c. Deve ler o arquivo **agenda.csv** e adicionar todas as informações em uma lista de contatos (banco_contatos). DICA (use o readlines()) e opção de arquivo **r+.**
   d. Crie uma função de **inserir** contado que insere as seguintes informações: **cpf, nome, sobrenome, email, telefone, curso, data_nasc, observacao.**
      i. Essa função deve verificar se o cpf que está sendo inserido já existe ou não dentro do banco_contatos. Se já existir o CPF não deve ser inserido.
      ii. Monte todos os dados em uma string com separador ; antes de inserir na lista de banco_contatos.
      iii. Após inserir o dado deve-se salvar(Use a função **salvar()**) o arquivo **agenda.csv**
   e. Crie uma função **atualizar contato** para atualizar as informações de um contato.
      i. Peça pro usuário digitar um CPF, verifique se existe. Se existir colete do usuário as novas informações e atualize o dado na lista. Para isso basta remover o usuário atual (Use a função remover(cpf)) e inserir logo após.
      ii. Após atualizar o dado deve-se salvar(Use a função **salvar()**) o arquivo **agenda.csv**

f. Crie uma função **mostrar lista** que percorre a lista banco_contatos, formata (retira o \n do final dos arquivos e quebra a lista com split()) e imprime os dados formatados de todos os usuários.
    i. Exemplo: Lista de contatos: CPF: 123, nome: Mário… Dica: Use o format da print.
g. Crie uma função **buscar contato por cpf** que recebe um cpf como parâmetro e verificar se o mesmo está na lista banco_contatos. Se estiver, imprimir os dados do contato buscado.
h. Crie uma função **buscar contato por email** que recebe um email como parâmetro e verificar se o mesmo está na lista banco_contatos. Se estiver, imprimir os dados do contato buscado.
i. Crie uma função **buscar contato por nome** que recebe um nome como parâmetro e verificar se o mesmo está na lista banco_contatos. Se estiver, imprimir os dados do contato buscado. DICA: Use a função lower() da string para não ter o erro de nomes com letras maiúsculas e minúsculas para essa busca.
j. Crie uma função **buscar contato por curso** que recebe um curso como parâmetro e verificar se o mesmo está na lista banco_contatos. Se estiver, imprimir os dados do contato buscado. DICA: Use a função lower() da string para não ter o erro de nomes com letras maiúsculas e minúsculas para essa busca.
k. Crie uma função **quantidade de contatos** que retorna a quantidade de contatos cadastrados na agenda e imprima na tela o valor.
l. Crie uma função **deletar contato por cpf** que recebe como parâmetro o cpf de um contato e o elimina da lista.
    i. Antes de deletar o contato da lista verifique se o mesmo existe na lista.
    ii. Após deletar o contato salve o arquivo de **agenda.csv** com a função **salvar()**
m. Crie uma função **deletar contato por email** que recebe como parâmetro o email de um contato e o elimina da lista.
    i. Antes de deletar o contato da lista verifique se o mesmo existe na lista.
    ii. Após deletar o contato salve o arquivo de **agenda.csv** com a função **salvar()**
n. Crie uma função **salvar**, responsável por salvar o arquivo **agenda.csv.**
o. Considere sempre colocar funções print nós métodos para indicar as opções que foram feitas.
p. Considere usar **try except para as operações de escrita e leitura do arquivo, por se tratar de uma opção que pode dar erro.**

3. **Módulo main:**
    a. Módulo que deve fazer a ligação dos demais módulos e fazer o controle das opções digitadas pelo usuário.
    b. Esse módulo deve importar os módulos menu.py e banco.py

c. Esse módulo deve tratar a opção inválida caso digitada
d. Só deve parar de mostrar o menu quando o usuário digitar a opção de sair.

4. **DICAS E BOAS PRÁTICAS**
    a. **Usem nomes de variáveis e funções de acordo com o que elas fazem**
    b. **Usem a criatividade para implementar mais funções ou melhorias**

OBS: O trabalho deve ser entregue em um **repositório NOVO** do GitHub chamado **agenda-de-contatos-python.** Crie um novo repositório, faça o clone, escreva o código e faça os commits à medida que o trabalho for evoluindo.

Materiais de apoio:

Métodos split, lower e replace da str:
https://www.w3schools.com/python/python_strings_methods.asp