Documentação do código em python:

Funcionalidades:

Este código permite ao usuário gerenciar uma lista de clientes e contar com estas funcionalidades:

Adicionar novo cliente;

Buscar cliente pelo nome;

Buscar cliente pelo código;

Ver a lista com todos os clientes

o código utilizar de funções do módulo "clientes.py" esse módulo que nós permite acessar as diversas funções do programa.

Requisitos para um bom funcionamento:

Requer a utilização do módulo no mesmo diretório do arquivo principal "main.py".

Deve-se fazer a importação das funções do módulo clientes para dentro do main, para que elas possam ser utilizadas.

Como Utilizar:

Para a utilização desse código execute-o em um ambiente python, e siga as instruções presentes no menu do programa.

caso tenha dificuldade aqui está a sequência de instruções do programa:

Para adicionar um novo cliente, selecione a 1 opção e em seguida siga os passos para fornecer as informações;

Para buscar um cliente por nome, selecione a 2 opção e forneça o nome a ser buscado;

Para buscar um cliente pelo código, selecione a 3 opção e forneça o código a ser buscado;

Para ver a lista de todos os clientes cadastrados, basta selecionar a opção 4;

Para finalizar o programa, selecione a opção 5;

observações:

Todos os dados dos clientes cadastrados ficam salvos em um "arquivo.txt", assim quando vier a executar o programa novamente esses dados já estarão lá.

Fazemos o uso de uma classe chamada cliente, ela quem armazena os dados dos clientes, contém os atributos: nome, endereço e codCliente. Usamos o método '__str__' para retornar uma representação em string do cliente

Funções utilizadas:

menu(): função que não retorna e nem recebe nada como parâmetro, apenas exibe o menu com as opções.

Preenche(lista):recebe uma lista como parâmetro, e retorna os dados dos clientes.

escreverArquivo(cliente): recebe os dados dos clientes e os salva em um "arquivo.txt".

lerArquivo(): Não recebe nada como parâmetro e retorna a lista com todos os clientes salvos no "arquivo.txt".

vereficarID(lista): recebe a lista de clientes como parâmetro e verifica se o código informado já existe, se sim, pede para digitar um código válido.

int_input(msg): função que vai forçar o usuário a digitar somente números inteiros, utilizando o try except.

str_input(msg): função semelhante a int_input(msg), mas aqui o usuário será forçado a informar dados do tipo string.

Busca_interpolacao_nomes(lista_nomes, chave): principal função do programa onde receber a lista completa, e uma variável chamada 'chave' com o nome a ser buscado. Dentro dessa função existe o algoritmo de busca por interpolação, onde esse algoritmo faz cálculos e chuta uma possível posição para este nome até encontrá-lo.

Ilustração do algoritmo:

```python
def busca_interpolacao_nomes(lista_nomes, chave):
    lista_clientes = lista_nomes
    lista_nomes = [cliente.nome for cliente in lista_clientes]
    lista_nomes.sort()
    inicio = 0
    fim = len(lista_nomes) - 1
    while inicio <= fim:
        posicao = inicio + int(((ord(chave[0]) - ord(lista_nomes[inicio][0])) / (
            ord(lista_nomes[fim][0]) - ord(lista_nomes[inicio][0]))) * (fim - inicio))
        if lista_nomes[posicao] == chave:
            return posicao
        elif lista_nomes[posicao] < chave:
            inicio = posicao + 1
        else:
            fim = posicao - 1
    return -1
```

busca_interpolacao_codigos(lista_codigos, chave): mesma funcionalidade do busca_interpolacao_nomes, porém ele faz a busca por código, agora a variável 'chave' fornece o código do cliente.

Ilustração do algoritmo:

```python
def busca_interpolacao_codigos(lista_codigos, chave):
    lista_clientes = lista_codigos
    lista_codigos = [cliente.codCliente for cliente in lista_clientes]
    lista_codigos.sort()
    inicio = 0
    fim = len(lista_codigos) - 1
    while inicio <= fim and lista_codigos[inicio] <= chave <= lista_codigos[fim]:
        posicao = inicio + int(((chave - lista_codigos[inicio]) / (
            lista_codigos[fim] - lista_codigos[inicio])) * (fim - inicio))
        if lista_codigos[posicao] == chave:
            return posicao
        elif lista_codigos[posicao] < chave:
            inicio = posicao + 1
        else:
            fim = posicao - 1
    return -1
```

Bibliotecas utilizadas:

Utilizamos a biblioteca time para vermos o tempo de execução do algoritmo de busca por interpolação.

Variáveis usadas no código:

clientes.py:

1. nome (na função preenche): Armazena o nome do cliente. É usado para criar um novo objeto Cliente.
2. endereco (na função preenche): Armazena o endereço do cliente. É usado para criar um novo objeto Cliente.

3. codCliente (na função preenche): Armazena o código do cliente. É usado para criar um novo objeto Cliente.

4. cliente (na função preenche): É um objeto da classe Cliente que armazena o nome, endereço e código do cliente. É adicionado à lista de clientes e escrito no arquivo de clientes.

5. lista (na função preenche): É uma lista de objetos Cliente. É usada para armazenar todos os clientes.

6. id (na função verificarID): Armazena o ID do cliente. É usado para verificar se o ID já existe na lista de clientes.

7. num (na função int_input): Armazena o número inteiro fornecido pelo usuário. É usado para retornar o número inteiro.

8. lista_nomes e lista_codigos (na função busca_interpolacao_nomes e busca_interpolacao_codigos): São listas de nomes e códigos de clientes, respectivamente. São usadas para realizar a busca por interpolação.

9. chave (na função busca_interpolacao_nomes e busca_interpolacao_codigos): É o valor que está sendo procurado nas listas de nomes e códigos.

10. posicao (na função busca_interpolacao_nomes e busca_interpolacao_codigos): É a posição calculada na lista durante a busca por interpolação.

11. inicio e fim (na função busca_interpolacao_nomes e busca_interpolacao_codigos): São os índices de início e fim usados durante a busca por interpolação.

12. lista_clientes (na função lerArquivo): É uma lista de objetos Cliente. É usada para armazenar todos os clientes lidos do arquivo.

13. linhas (na função lerArquivo): É uma lista de todas as linhas no arquivo de clientes. É usada para ler os dados dos clientes do arquivo.

14. linha (na função lerArquivo): É uma linha individual do arquivo de clientes. É usada para extrair os dados do cliente.

15. arquivo (na função lerArquivo e escreverArquivo): É o arquivo de clientes. É usado para ler e escrever os dados dos clientes.

16. msg (na função int_input e str_input): É a mensagem que é exibida ao solicitar a entrada do usuário.

main.py:

1. lista: Esta é uma lista de clientes. É preenchida pela função lerArquivo(), que presumivelmente lê um arquivo e retorna uma lista de objetos Cliente.
2. opc: Esta variável armazena a opção escolhida pelo usuário no menu. A função int_input() é usada para obter a entrada do usuário.
3. cliente: Esta variável é usada para armazenar um novo cliente que é adicionado à lista quando o usuário escolhe a opção 1 no menu. A função preenche() é usada para criar o novo cliente.
4. nome_busca: Esta variável armazena o nome que o usuário deseja buscar na lista de clientes quando a opção 2 é escolhida.
5. resultado: Esta variável armazena o resultado da busca por nome ou código. Se o nome ou código for encontrado, resultado armazenará o índice do cliente na lista; caso contrário, será -1.
6. cod_busca: Esta variável armazena o código que o usuário deseja buscar na lista de clientes quando a opção 3 é escolhida.