

Computação I

Lista de exercícios 7 – Interação com o usuário

Atenção! Leia as instruções antes de fazer a lista! - Data de entrega: 09/11/2022

Por favor, não deixe de respeitar a padronização do nome do arquivo e dos nomes das funções, pois isso nos ajuda na correção.

A menos que esteja explicitamente pedido na questão, não utilize nenhum método ou função já existente do Python exceto pelo input, print, len, range, input, format, upper e lower (de forma simplificada, função é tudo o que precisa ser chamado com parênteses, e método é tudo o que precisa ser chamado com parênteses, mas que está associado a uma variável, isto é, variavel.metodo(args), por exemplo, o len é uma função e o upper e o lower são métodos). **Apenas as funções de transformação de tipo int e float são permitidas nessa lista.** Não é necessário testar se os dados passados através do input são válidos (considere que os valores passados por input podem ser convertidos para inteiro, quando necessário, ou float, quando necessário). **Atenção para a diferença entre o return e o print.** O return permite que o valor seja utilizado fora da função, enquanto o print só imprime na tela. Somente a função principal do seu código deve terminar com o número 1.

Para ajudar na correção, é muito importante verificar, no arquivo de testes, os exemplos de funcionamento do código. Sua função deve funcionar exatamente como os exemplos. Não adicione nenhum input além dos pedidos. Além disso, **todas as mensagens a serem impressas na tela devem seguir exatamente o modelo especificado nos exemplos. Isto é, use exatamente as mesmas palavras e caracteres para imprimir as mensagens pedidas nas questões, e não imprima mais mensagens do que o especificado.** Por exemplo: se a questão diz para imprimir 'Nao ha clientes cadastrados', imprima exatamente esta frase, e não imprima frases semelhantes no lugar, como 'Nenhum cliente cadastrado' ou 'Não há clientes cadastrados'.

1. Desejamos fazer um programa que ajude uma clínica veterinária a organizar os dados dos clientes. O programa deve imprimir um menu com seis opções: (1) inserir os dados de um ou mais clientes, (2) mostrar a lista de clientes, (3) listar todos os clientes de uma espécie determinada, (4) listar todos os animais de um cliente, (5) remover os dados de um ou mais clientes, ou (6) para terminar. Todas as opções, exceto a de terminar o programa, devem, **obrigatoriamente**, ser implementadas em funções separadas, uma para cada opção. Não utilize variáveis globais. A estrutura de dados que guarda as informações de clientes dentro do programa é uma tupla de tuplas onde cada subtupla contém o nome do dono do animal (string), o nome do animal (string), a espécie (string), a idade (inteiro), e o peso do animal (float). **Atenção! Os tipos indicados precisam ser respeitados: tanto a tupla de tuplas quanto os tipos dentro das subtuplas! A manipulação da tupla de tuplas e dos tipos de dados fazem parte do problema. Não use nenhuma variável do tipo "lista" ou "dict" em nenhuma parte do programa!** O programa deve ficar em loop até que o usuário digite 6. O funcionamento de cada opção do menu é descrito abaixo.

Opção 1 do menu: seu programa deve pedir para ele digitar o nome do cliente, o nome do animal, a espécie do animal, a idade do animal, e o peso do animal, utilizando inputs separados para cada informação. Então, ele deve criar uma tupla com essas informações, a qual deve ser inserida na tupla com todos os clientes; e perguntar se o usuário deseja inserir outro cliente. Se a resposta for 's', seu programa deve continuar repetindo os mesmos passos anteriores, até que o usuário digite 'n' como resposta da última pergunta. Se o usuário digitar 'n', seu programa deve voltar ao menu principal.

Opção 2 do menu: seu programa deve mostrar cada cliente que está na tupla de tuplas, um cliente por linha, indicando o nome do cliente, o nome do animal, a espécie, a idade e o peso do animal. A formatação é importante, não imprimia diretamente a tupla ou a subtupla (veja o exemplo abaixo). Se não houver nenhum cliente cadastrado, imprima 'Nao ha clientes cadastrados'. Além disso, o peso do animal deve ser sempre formatado com duas casas decimais.

Exemplo de como as informações devem ser exibidas na tela na opção 2:

Dono: Fulano ; Animal: Rex ; Espécie: Cachorro ; Idade: 4 ano(s) ; Peso: 4.23 kg

Dono: Beltrano ; Animal: Bartos ; Espécie: Gato ; Idade: 3 ano(s) ; Peso: 3.50 kg

Opção 3 do menu: seu programa deve perguntar a espécie desejada, buscar a mesma na tupla de tuplas, e imprimir as informações de todos os clientes que possuem um animal daquela espécie, um cliente por linha, mostrando o nome do cliente, o nome do animal, a sua idade e o seu peso (a espécie não é mostrada aqui, já que ela foi usada na busca – veja o exemplo abaixo). O peso deve estar formatado com duas casas decimais. Letras maiúsculas e minúsculas não devem influenciar na busca. Se a espécie não foi encontrada na busca, imprima 'Espécie nao encontrada'. Se não houver clientes cadastrados, imprima 'Nao ha clientes cadastrados.'

Exemplo de como as informações devem ser exibidas na tela na opção 3:

Cliente: Fulano ; Animal: Rex ; Idade: 4 ano(s) ; Peso: 4.23 kg

Cliente: Cicrano ; Animal: Joey ; Idade: 6 ano(s) ; Peso 3.80 kg

Opção 4 do menu: seu programa deve perguntar o nome do cliente desejado, buscar o mesmo na tupla de tuplas, e imprimir as informações de todos os animais daquele cliente, um por linha, mostrando o nome do animal, a sua espécie e a sua idade (o cliente não é mostrado aqui, já que ele foi usado na busca – veja o exemplo abaixo). Se o cliente não foi encontrado na busca, imprima 'Cliente nao encontrado'. Letras maiúsculas e minúsculas não devem influenciar na busca.

Exemplo de como as informações devem ser exibidas na tela na opção 4:

Animal: Joey ; Espécie: Cachorro ; Idade: 6 ano(s) ; Peso: 3.80 kg

Animal: Biriba ; Espécie: Gato ; Idade: 2 ano(s) ; Peso: 2.50 kg

Opção 5 do menu: seu programa deve perguntar o nome de um cliente, buscar o mesmo na tupla de tuplas, e exibir uma mensagem informando quantos clientes foram encontrados, no seguinte formato: 'X cliente(s) foram encontrados' (onde X representa o número de registros encontrados na busca). Então, o programa deve imprimir as informações do primeiro cliente encontrado, e perguntar ao usuário se ele deseja confirmar a exclusão. Se o usuário digitar 's', então o cliente deve ter seu registro excluído, e o programa deve perguntar se o usuário deseja excluir outro cliente. Caso contrário, o registro deve ser mantido, e o programa deve imprimir as informações do próximo cliente encontrado. Se, após exibir todos os registros encontrados, o usuário decidir não excluir nenhum deles, o programa deve perguntar se o usuário deseja excluir outro cliente. Se o usuário digitar 's' quando for perguntado pelo programa se deseja excluir outro cliente, então o programa deve repetir os passos definidos anteriormente. Caso contrário, o programa deve retornar para o menu principal. Se o cliente não for encontrado na tupla de tuplas, o programa deve imprimir a mensagem 'Cliente nao encontrado' e retornar ao menu principal. Veja o quinto, sexto e sétimo exemplos do arquivo de testes para atentar a como a função de exclusão deve operar.

Exemplo de como o registro de um cliente deve ser exibido antes de confirmar a exclusão:

1 cliente(s) encontrado.

Cliente: Fulano ; Animal: Rex ; Espécie: Cachorro ; Idade: 4 ano(s) ; Peso: 4.23 kg

A função principal, que deve ser chamada para que o programa dessa questão seja executado, pode ter qualquer nome, porém deve necessariamente terminar com o número 1 (por exemplo: principal1(), main1(), questao1()...). Ela deve retornar a tupla de tuplas que foi criada. Veja os exemplos de execução no arquivo de testes. Nas execuções mostradas, a última linha é o valor de retorno da função principal, a qual é chamada com o print (por exemplo, print(main1())).