

Nome: _____

Matrícula: _____

A prova é individual e deverá ser entregue até o horário da aula na próxima Terça-Feira (10/05/2021). As questões abaixo referem-se ao código Python contido no arquivo da prova. Sim, eu sei que as questões estão somando 12 pontos, mas a última questão é mais desafiadora. Se o aluno completar os 12 pontos, os 2 pontos extras serão contabilizados no projeto.

1. (2 pontos) Modifique o código da agência para perguntar qual operação o usuário quer fazer. Se for **depositar** ou **sacar**, peça o tipo e o número da conta, para encontrá-la na coleção da agência, e peça o valor para realizar a operação. Se for **criar**, reuse o código que pede as informações, cria e insere a conta na coleção da agência. Se for **fechar**, peça as informações necessárias para remover a conta da coleção da agência. Imprima a conta ao terminar a operação.
2. (2 pontos) Quais erros podem acontecer quando o usuário digitar o valor do saldo inicial da conta? Trate esses erros.
3. (2 pontos) Ao rodar o script Python, leia o arquivo *agencia.txt* e use seu conteúdo para inicializar as coleções de contas e poupanças da agência. Note que isso pode resultar em erro, caso o arquivo não exista. Trate o erro, para evitar que o programa quebre. Dica: ver Seção 1.6.5.1 do material.
4. (2 pontos) Modifique a conta para que ela guarde uma coleção das suas transações (depósitos e saques) e adicione um método *imprimir_extrato*, que imprime os dados das transações.
5. (4 pontos) Da forma como o código se encontra, será necessário copiar e colar muito código para que o banco possa ter múltiplas agências. Crie a classe agência, lembrando que ela vai ter que ter métodos que permitem criar e inserir contas e realizar operações sobre elas, de imprimir o arquivo com os dados das contas ao final da execução e de ler o arquivo no início. O arquivo *main.py* vai passar a representar o comportamento de um banco, portanto ele deverá permitir ao usuário criar agências e realizar operações sobre as contas que pertencem a elas, além de imprimir os arquivos de todas as agências ao final da execução e ler os arquivos de todas as agências no início. Dica: para listar todos os arquivos em um diretório, use a função *listdir* da biblioteca *os*.