

Exercício de Fixação – Noções de Orientação a Objetos em Python
--

1. Implemente uma classe chamada Data, que poderá ser usada para representar uma data. A classe deverá ter:
 - a) Os atributos privados: dia, mês e ano;
 - b) Construtor;
 - c) Métodos acessadores e modificadores;
 - d) Método `__str__(self)`, que deverá retornar a data no formato dd/mm/aaaa.

Escreva um programa para criar objetos dessa classe e testar os métodos.

2. Implemente uma classe chamada Aluno que deve ter:
 - a) Atributo matricula (número inteiro), nome (string), notas (*list*);
 - b) Construtor para inicializar todos os atributos;
 - c) Métodos acessadores para nome e matrícula. Este último deve retornar a matrícula como uma String formatada;
 - d) Método `media(self)` que retorna a media das notas;
 - e) Método alterador apenas para o nome;
 - f) Método `adiciona_nota(self, nota)`, para adicionar uma nota à lista de notas do aluno.

Escreva um programa para criar objetos dessa classe e testar os métodos.

3. Escreva uma classe que represente um país. Um país tem como atributos privados o seu nome, o nome da capital, sua dimensão em Km² e uma lista de países com os quais ele faz fronteira. Represente a classe e forneça os seguintes construtores e métodos:
 - a) Construtor que inicialize o nome, capital e a dimensão do país;
 - b) Métodos de acesso para os atributos indicados no item (a);
 - c) Método que retorne a lista de países que fazem fronteira;
 - d) Método que adiciona o nome de país a lista de fronteiras (verificar se o nome já existe na lista).
 - e) Método `__str__(self)`.

Escreva um programa para criar objetos dessa classe e testar os métodos.

4. Implemente uma classe `Conta_Corrente`, cujos atributos dos seus objetos são **numero**, **saldo** e **nome_titular**. Ainda na classe, implemente os métodos depositar e sacar, parametrizando neles o valor a ser depositado ou sacado, respectivamente. O método depositar não possui retorno, devendo apenas incrementar o valor do saldo. O **método sacar** deverá **retornar** um valor **booleano** (**true** – se sacou com sucesso, pois há saldo suficiente, decrementando-o; ou **false** caso contrário). A classe também deverá possuir um construtor, para inicializar os valores dos atributos dos objetos instanciados. Em seguida:

a) Escreva um programa para criar dez instâncias (objetos) de `Conta_Corrente`, armazenando-os em uma *list*. Os valores para inicialização dos objetos deverão ser lidos do teclado;

b) Criados os objetos, disponibilize um menu de operações para o usuário. **Em um loop**, o programa ficará solicitando ao usuário, qual o número da operação abaixo ele deseja realizar:

1) Depositar

Ao digitar a opção 1, o programa deverá ler o número da conta e o valor a ser depositado e realizar a operação.

2) Sacar

Ao digitar a opção 2, o programa deverá ler o número da conta e o valor a ser sacado e realizar a operação, testando se o saque ocorreu ou não.

3) Saldo

Ao digitar a opção 3, o programa deverá ler o número da conta e exibir o seu saldo.

4) Sair

Ao digitar a opção 4, o programa deverá ser encerrado. O loop só será encerrado quando esta operação for informada.