

Introdução

Nesta aula, como exemplo de definição de classes, vamos criar classes com o intuito de modelar um banco de varejo como o Banco do Brasil, Itaú etc. Vamos definir classes para representar clientes do banco e para representar contas correntes dos clientes. E futuramente poderemos definir uma classe para modelar o próprio banco. As duas classes de hoje serão denominadas *Cliente* e *Conta*. A classe *Cliente* será muito simples: conterá apenas os atributos *nome* e *telefone* para identificar um cliente e o método construtor. A classe *Conta* conterá os atributos *saldo*, *clientes* (uma conta pode ter mais de um correntista, considere o caso das contas conjuntas), *numero da conta* e as *operacoes realizadas na conta*. Além disso, a classe *Conta* conterá os métodos: `__init__` para criar um novo objeto da classe, *resumo* para imprimir um resumo da conta, *saque* para realizar uma operação de saque na conta, *deposito* para realizar uma operação de depósito e, finalmente, o método *extrato* para imprimir um extrato da conta listando todas as operações realizadas na conta até o momento. O programa deverá definir também uma função *main* para testar nossas definições das classes *Cliente* e *Conta*. A título de ajuda, mostraremos, a seguir, uma sugestão de implementação para os métodos *deposito* e *extrato*:

```
class Conta:
    def __init__(self, clientes, numero, saldo=0.00):
        ...
        self.operacoes = []
        self.deposito(saldo)

    ...

    def deposito(self, valor):
        self.saldo += valor
        self.operacoes.append(("DEPÓSITO", valor))

    def extrato(self):
        print("\nExtrato da CC nº %s\n" % self.numero)
        for op in self.operacoes:
            print("%10s %10.2f\n" % (op[0], op[1]))
        print("      SALDO %10.2f" % self.saldo)
```

Siga as instruções abaixo para elaborar todo o programa.

Instruções

1. Abra o IDLE e crie um novo arquivo fonte denominado `p13.py`. Não se esqueça de salvá-lo de tempos em tempos, porque pode ocorrer uma falha de energia elétrica durante a aula prática.
2. Digite os comentários obrigatórios (nome, matrícula, data e uma breve descrição sobre o que o programa faz).
3. Organize seu programa com as classes *Cliente* e *Conta* e a função `main()`.
4. A classe *Cliente* terá apenas o construtor onde serão definidos os atributos *nome* e *telefone* cujos valores iniciais serão passados como parâmetros.
5. A classe *Conta* deve definir os atributos *clientes*, *numero da conta*, *saldo*, *operacoes* e realizar o depósito inicial que terá como valor o saldo inicial passado como parâmetro,

Prática 13 – INF101 – 2018/II – 2 pontos

além dos valores iniciais dos outros atributos também, exceto operacoes. Implemente, na classe Conta, os métodos: saque que deve ter o valor do saque passado como parâmetro, deposito que deve ter o valor do depósito passado como parâmetro, resumo que não precisa de parâmetro, e, finalmente, extrato (veja sugestão acima na introdução deste roteiro). **Observações:** Para realizar um saque, o valor do saque não pode ser maior que o saldo atual da conta. Os métodos saque e deposito devem registrar as respectivas operações na lista operacoes. Veja a sugestão dada acima na introdução. O método resumo imprime apenas o número da conta corrente e seu saldo. Veja o formato da impressão no exemplo de execução do programa ilustrado na próxima página.

6. A função main deve testar as classes Cliente e Conta. Use os dados que são mostrados a seguir:

```
cliente1 = Cliente("João Silva", "3234-7890")
cliente2 = Cliente("Maria Silva", "3234-7890")
cliente3 = Cliente("José Vargas", "2567-0987")

conta1 = Conta([cliente1, cliente2], 1, 1000.00)
conta2 = Conta([cliente3], 2, 500.00)

conta1.saque(50.00)
conta2.deposito(300.00)
conta1.saque(190.00)
conta2.deposito(95.15)
conta2.saque(256.71)

conta1.resumo()
conta2.resumo()

conta1.extrato()
conta2.extrato()
```

7. Não se esqueça de chamar a função main no final de seu código fonte para iniciar todo o processo.
8. Veja como deve ser a saída do programa no exemplo dado na próxima página.

Após certificar-se de que seu programa esteja correto, envie o arquivo do programa fonte (p13.py) através do sistema de entrega do LBI.

Exemplo de Execução do Programa

CC nº 1 Saldo: 760.00

CC nº 2 Saldo: 638.44

Extrato da CC nº 1

DEPÓSITO 1000.00

SAQUE 50.00

SAQUE 190.00

SALDO 760.00

Extrato da CC nº 2

DEPÓSITO 500.00

DEPÓSITO 300.00

DEPÓSITO 95.15

SAQUE 256.71

SALDO 638.44