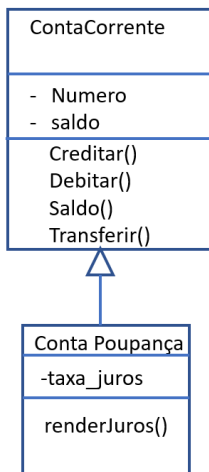


IFPI

CURSO: TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA: PROGRAMAÇÃO ORIENTADA À OBJETOS

PROF.: ROGÉRIO BATISTA



1. Implemente a hierarquia de classes descrita no diagrama acima. Defina os construtores das classes (`__init__`) e a saída de dados (`__str__`) nas classes indicadas. Implemente os métodos `creditar(...)`, `debitar(...)`, `transferir(...)` e `saldo()` da classe `ContaCorrente`. O método `creditar(...)` adiciona um valor recebido como parâmetro e adiciona ao atributo `saldo`. O método `debitar(...)` subtrai do atributo `saldo` o valor passado como parâmetro, somente se este valor for menor ou igual ao saldo da conta. O método `saldo()` na verdade é um decorador `getter`. O método `transferir(...)` transfere um valor da conta de origem para outra conta (verificar se a conta de origem tem saldo suficiente. Passar como parâmetros o valor e um objeto conta que é a conta destino). O método `renderJuros(...)` da subclasse `ContaPoupança` aplica o percentual definido no parâmetro `taxa_juros` em cima do atributo `saldo` da classe `ContaCorrente`, atualizando-o. Crie uma classe `ContaImposto` que herda de `ContaCorrente` e possui um atributo `percentualImposto`. Esta classe também possui um método `calculaImposto()` que subtrai do saldo, o valor do próprio saldo multiplicado pelo percentual do imposto. Faça a classe de teste e execute. Crie pelo menos 2 instancias de cada classe e mostre os valores dos seus atributos, teste todos os métodos e mostre ao final os valores dos atributos de todos os objetos criados através do comando: `print(<objeto>)`. Isto só será possível se o método `__str__()` for implementado nas classes.