



UNIVERSIDADE FEDERAL DO CEARÁ
Departamento de Computação

Prof. Lincoln Souza Rocha

CK0442- Técnicas de Programação para Ciência de Dados - T02

TP03
2024.2

1. Implementando a classe ContaPoupanca via Herança

Descrição: Altere o módulo `sisbanco.py` (criado no TP02) para incluir a classe `ContaPoupanca` seguindo a especificação abaixo. (**Dica.** A Nota de Aula 06 descreve a implementação completa da classe `ContaPoupanca`).

```
class ContaPoupanca(Conta):  
    def __init__(self, numero: str):  
        pass  
  
    def render_juros(self, taxa: float) -> None:  
        pass
```

2. Implementando a classe ContaEspecial via Herança

Descrição: Altere o módulo `sisbanco.py` (criado no TP02) para incluir a classe `ContaEspecial` seguindo a especificação abaixo. Lembre-se que a classe `ContaEspecial` deve redefinir (sobrescrever) o método `creditar()` da classe `Conta` para garantir o incremento do bônus. (**Dica.** A Nota de Aula 06 descreve a implementação completa da classe `ContaEspecial`).

```
class ContaEspecial(Conta):  
    def __init__(self, numero: str):  
        pass  
  
    def render_bonus(self) -> None:  
        pass
```

3. Rendendo Juros no SisBanco

Descrição: No módulo `sisbanco.py` (criado no TP02), modifique a classe `Banco` para que seja possível render juros de uma conta poupança. Isto é, adicione um novo método que rende os juros em contas poupança (veja descrição abaixo). A taxa de correção da poupança do Banco deve ser um atributo privado da classe `Banco`. Além disso, crie métodos para alterar e recuperar o valor da taxa de correção da poupança (veja a descrição abaixo). (**Dica.** Apenas objetos de classe `ContaPoupanca` podem render juros, use `isinstance()` para verificar o tipo dinâmico dos objetos).

```
class Banco:  
    def __init__(self, taxa: float):  
        pass  
    (...)   
    def render_juros(self, numero: str) -> None:  
        pass  
  
    def get_taxa(self) -> float:  
        pass  
  
    def set_taxa(self, taxa: float) -> None:  
        pass
```

4. Rendendo Bônus no SisBanco

Descrição: No módulo `sisbanco.py` (criado no TP02), modifique a classe `Banco` para que seja possível render bônus em uma conta especial. Isto é, adicione um novo método que rende bônus em contas especiais (veja descrição abaixo). (**Dica.** Apenas objetos da classe `ContaEspecial` podem render bônus, use `isinstance()` para verificar o tipo dinâmico dos objetos).

```
class Banco:
    def __init__(self, taxa: float):
        pass
    (...)
    def render_bonus(self, numero: str) -> None:
        pass
```

5. Interagindo com o Sistema Bancário

Descrição: Altere o módulo `terminal_atendimento.py` (criado no TP02) para trabalhar com as novas classes e funcionalidades do `SisBanco`. A descrição abaixo fornece os detalhes das novas funcionalidades que devem ser implementadas (i.e., opções 5, 6 e 7 do menu principal).

```
def terminal():
    sisbanco = Banco()
    while(True):
        print("SisBanco::Bem-Vindo!")
        print(" :: Opcoes:: ")
        print("[0] - Cadastrar_Conta")
        print("[1] - Creditar")
        print("[2] - Debitar")
        print("[3] - Transferir")
        print("[4] - Consultar_Saldo")
        print("[5] - Render_Juros")
        print("[6] - Render_Bonus")
        print("[7] - Alterar_Taxa_Juros")
        print("[8] - Sair")
        opcao = input("Digite: ")

        if opcao == 0:
            #qual tipo de conta a ser criada:
            #S - Simples | P - Poupanca | E - Especial
            #solicite o numero da conta a ser criada
            #instancie uma conta do tipo selecionado com esse numero
            #cadastre a conta no sisbanco

        elif opcao == 1:
            #solicite o numero da conta alvo
            #solicite o valor a ser creditado
            #realize a operacao de credito no sisbanco

        elif opcao == 2:
            #solicite o numero da conta alvo
            #solicite o valor a ser debitado
            #realize a operacao de debito no sisbanco

        elif opcao == 3:
            #solicite o numero da conta origem
            #solicite o numero da conta destino
            #solicite o valor a ser transferido
            #realize a operacao de transferencia no sisbanco
```

```
elif opcao == 4:
    #solicite o numero da conta alvo
    #realize a operacao de obtencao de saldo no sisbanco
    #exiba o saldo na tela

elif opcao == 5:
    #solicite o numero da conta alvo
    #realize a operacao correcao da poupanca no sisbanco

elif opcao == 6:
    #solicite o numero da conta alvo
    #realize a operacao conversao/rendimento de bonus no sisbanco

elif opcao == 7:
    #solicite a nova taxa de correcao da poupanca
    #realize a operacao de alteracao da taxa no sisbanco

elif opcao == 8:
    print("SisBanco::Bye!")
    break

if __name__ == "__main__":
    terminal()
```