

Desafio!

Dados da Conta Bancária

Crie um software similar aos utilizados em Bancos para gerenciamentos de Contas Correntes e Contas Poupanças.

Inicie pela criação dos Atributos.

```
Classe ContaBanco
// Atributos
publico numConta: Inteiro
protegido tipo: Caractere
privado dono: Caractere
privado saldo: Real
privado status: Logico
// Métodos
publico Metodo abrirConta()
(...)
FimMetodo
publico Metodo fecharConta()
(...)
FimMetodo
publico Metodo depositar()
(...)
FimMetodo
publico Metodo sacar()
(...)
FimMetodo
publico Metodo pagarMensal()
(...)
FimMetodo
FimClasse
```

Crie o Construtor.

```
Classe ContaBanco
// Atributos
publico numConta: Inteiro
protegido tipo: Caractere
privado dono: Caractere
privado saldo: Real
privado status: Logico
// Métodos Especiais
publico Metodo Construtor()
    saldo = 0
    status = falso
FimMetodo
```

Desenvolva os Métodos especiais.

```
Classe ContaBanco
// Atributos
publico numConta: Inteiro
protegido tipo: Caractere
privado dono: Caractere
privado saldo: Real
privado status: Logico
// Métodos Especiais
publico Metodo setDono(d:Caractere)
    dono = d
FimMetodo
publico Metodo getDono()
    retorne dono
FimMetodo
publico Metodo setSaldo(s:Real)
    saldo = s
FimMetodo
publico Metodo getSaldo()
    retorne saldo
FimMetodo
publico Metodo setStatus(s:Logico)
    status = s
FimMetodo
publico Metodo getStatus()
    retorne status
FimMetodo
```

Desenvolva o Método necessário para abrir uma nova conta.

```
Classe ContaBanco
// Atributos
publico numConta: Inteiro
protegido tipo: Caractere
privado dono: Caractere
privado saldo: Real
privado status: Logico
// Métodos
publico Metodo abrirConta(t:Caractere)
    setTipo(t)
    setStatus(verdadeiro)
    se (t = "CC") entao
        saldo = 50
    senao se (t = "CP") entao
        saldo = 150
    FimSe
FimMetodo
publico Metodo fecharConta()
    (...)
FimMetodo
publico Metodo depositar()
    (...)
FimMetodo
publico Metodo sacar()
    (...)
FimMetodo
publico Metodo pagarMensal()
```

Desenvolva o Método necessário para encerrar uma conta.

```
publico Metodo abrirConta()  
    (...)  
FimMetodo  
publico Metodo fecharConta()  
    se (saldo > 0) entao  
        Escreva("Conta com dinheiro")  
    senao se (saldo < 0) entao  
        Escreva("Conta em débito")  
    senao  
        setStatus(false)  
    FimSe  
  
FimMetodo  
publico Metodo depositar()  
    (...)  
FimMetodo  
publico Metodo sacar()  
    (...)  
FimMetodo
```

Desenvolva o Método necessário para depósito de dinheiro.

```
publico Metodo abrirConta()  
    (...)  
FimMetodo  
publico Metodo fecharConta()  
    (...)  
FimMetodo  
publico Metodo depositar(v:Real)  
    se (status = verdadeiro) entao  
        setSaldo(getSaldo() + v)  
    senao  
        Escreva("Impossível depositar")  
    FimSe  
  
FimMetodo  
publico Metodo sacar()  
    (...)  
FimMetodo  
publico Metodo pagarMensal()  
    (...)  
FimMetodo
```

Desenvolva o Método necessário para realização de saques.

```
publico Metodo abrirConta()  
    (...)  
FimMetodo  
publico Metodo fecharConta()  
    (...)  
FimMetodo  
publico Metodo depositar()  
    (...)  
FimMetodo  
publico Metodo sacar(v:Real)  
    se (status = verdadeiro) entao  
        se (saldo > v) entao  
            saldo = saldo - v  
        senao  
            Escreva("Saldo insuficiente")  
        FimSe  
    senao  
        Escreva("Impossível sacar")  
    FimSe  
  
FimMetodo
```

Desenvolva o Método necessário para pagamento de tarifa mensal.

```
publico Metodo sacar()  
    (...)  
FimMetodo  
publico Metodo pagarMensal()  
    var v: Real  
    se (tipo = "CC") entao  
        v = 12  
    senao se (tipo = "CP") entao  
        v = 20  
    fimse  
    se (status = verdadeiro) entao  
        se (saldo > v) então  
            saldo = saldo - v  
        senao  
            Escreva("Saldo insuficiente")  
        fimse  
    senao  
        Escreva("Impossível pagar")  
    fimse  
FimMetodo
```