

Lista de Exercícios 3

1. Complete a classe ponto com operações aritméticas e de comparação.
2. Crie a classe matriz com com operações aritméticas e de comparação.
3. Faça um programa que calcule a quantidade de azulejos quadrados que serão necessários para cobrir cada uma das paredes, chão e teto de um banheiro que possui duas aberturas retangulares (uma porta e uma janela). A dimensão do azulejo, a altura, comprimento e largura do banheiro, as alturas e larguras da porta e da janela serão fornecidos.

Elementos conceituais:

- ✓ Paredes, teto, chão, porta e janela: retângulos
- ✓ Azulejo: quadrado

Classe quadrado:

Atributos:

Tamanho do lado

Métodos:

Alterar/Consultar valor do lado,

Calcular área

Calcular perímetro

Exibir

Classe retângulo:

Atributos:

Base, Altura

Métodos:

Alterar/Consultar atributos,

Calcular área

Calcular perímetro

Exibir

4. Criar a classe círculo com métodos para:
 - a) calcular a área
 - b) calcular o perímetro
 - c) verificar se um dado ponto está dentro do círculo
 - d) clonar o círculo
 - e) exibir
 - f) alterar o centro ou alterar somente o x ou somente o y
 - g) alterar o raio
 - h) retornar o centro ou somente o x do centro ou somente o y
 - i) retornar o raio
 - j) verificar a distância do y de um ponto em relação ao centro
 - k) verificar a distância do x de um ponto em relação ao centro

Faça um programa que determine a posição relativa entre duas circunferências (tangentes, secantes, externas, internas ou concêntricas) cujos centro e raio são fornecidos pelo usuário.

5. Criar a classe Triângulo com o seguintes:

- a) Atributos: ptos A,B,C
- b) Métodos:
 - a) área
 - b) perímetro
 - c) baricentro
 - d) Comprimento dos lados
 - e) medida dos ângulos
 - f) alterar os pontos A,B e C
 - g) Exibir triângulo

6. Crie a seguinte classe conta bancária:

ContaBancaria
+agencia: integer +codigo: string +dataDeCriacao: string +senha: string +saldo: float +cancelada: boolean +retirar(quantia:float) +depositar(quantia:float) +obterSaldo()

7. Crie a seguinte classe pessoa:

Pessoa
+codigo: integer +nome: string +altura: integer +idade: integer +nascimento: date +escolaridade: string +salario: float +crescer(centimetros:integer) +formar(titulo:string) +envelhecer()

Definição dos métodos Pessoa

- crescer - adicionar mais cm à altura atual
- formar - modificar escolaridade
- envelhecer - adicionar + 1 em idade
- construtor :
 - altura deverá ser inicialmente 0.3
 - idade e salário deverão ser 0
 - nascimento deverá ser o **dia e hora do momento**
 - código e nome deverão ser fornecidos como parâmetros

8. Demonstre que compreendeu bem as relações entre classes considerando a classe **Cliente** dada e criando duas novas classes como pedido abaixo. A classe **Cliente não pode ser alterada** e deve ser copiada no arquivo cliente.py:

```
class Cliente:
    def __init__(self, nome, idade, cpf):
        self.nome=nome
        self.idade = idade
        self.cpf = cpf
        self.valorTotalDeCompras=0.0
    def __str__(self):
        return ('{}-{}-{}-{}'.format(self.nome,
        self.idade,self.cpf,self.valorTotalDeCompras))
    def __repr__(self):
        return ('{}-{}-{}-{}'.format(self.nome,
        self.idade,self.cpf,self.valorTotalDeCompras))
    def novaCompra(self, valorCompra):
        self.valorTotalDeCompras+=valorCompra
    def informarValorTotalDeCompras(self):
        return self.valorTotalDeCompras
    def __lt__(self,outroCli):
        return self.nome < outroCli.nome
```



DEPARTAMENTO
DE INFORMÁTICA
PUC-RIO

INF1026 2017.2

Ciclo Básico do Centro Técnico Científico (CB-CTC)

E-mail: coordprog@inf.puc-rio.br

URL: www.inf.puc-rio.br/~inf1025

Escreva a classe **Loja**.

- ✓ Uma loja quando criada tem:
 - um nome de loja
 - um gerente
 - um cnpj (string)
 - uma lista de clientes (inicialmente vazia).
- ✓ Uma loja só é criada recebendo: um nome, um gerente e um cnpj
- ✓ Ao ser dado *print* em uma loja devem ser exibidos: seu cnpj, seu nome e o gerente
- ✓ Uma loja pode:
 - incluir um novo cliente em sua lista, recebendo para isso um cliente **criado** (método: ***incluirCliente***)
 - exibir os dados de todos os seus clientes (método: ***exibeClientes***)
 - calcular o valor total de gastos de seus clientes, retornando-o. (método: ***calculaValorTotal***)

Escreva a classe **ClientePreferencial**.

- ✓ Um *ClientePreferencial* é um **Cliente** que, além de tudo que um cliente tem, possui também um percentual de desconto a ser aplicado em suas compras.
- ✓ Obviamente um **ClientePreferencial** é uma subclasse (classe derivada) de **Cliente**, e portanto deve ser criada como tal e fazer o uso apropriado dos métodos da classe **Cliente**.
- ✓ Um cliente preferencial só é criado recebendo nome, idade, cpf e percentual de desconto (inteiro).
- ✓ Ao ser dado um *print* em um cliente preferencial é exibido tudo que um cliente tem mais o desconto.
- ✓ Um cliente ao fazer uma **novaCompra** de um certo valor, na verdade paga um valor real que é o valor da compra com o percentual de desconto aplicado. É o valor real que o cliente paga que deve ser acrescentado ao seu valorTotalDeCompras.

Abaixo um código para ser executado para testar as classes construídas por você.

```
c1=Cliente('Vik',34,'1234')
c2=Cliente('Ana',40,'4567')
c1.novaCompra(1500)
c2.novaCompra(2000)
c1.novaCompra(700)
print(c1)
print(c2)
lj = Loja('MAGAZINE','dudu','9999')
print(lj)
lj.incluirCliente(c1)
lj.incluirCliente(c2)
lj.incluirCliente(Cliente('Edu',55,'3456'))
print(lj)
lj.exibeClientes()
print('Total da loja: ',lj.calculaValorTotal())
cp= ClientePreferencial('GUGA',38,'9876',20)
print(cp)
cp.novaCompra(1000)
print(cp)
```

Resultado esperado:

```
numVik-34-1234-2200.0
Ana-40-4567-2000.0
CNPJ: 9999 Loja: MAGAZINE Gerente: dudu
CNPJ: 9999 Loja: MAGAZINE Gerente: dudu
[Vik-34-1234-2200.0, Ana-40-4567-2000.0, Edu-55-3456-0.0]
Total da loja: 4200.0
GUGA-38-9876-0.0-20
GUGA-38-9876--19000.0-20
```