1. Construa classe Circulo considerando que um círculo é representado pelo centro e raio. Portanto:

Atributos: centro (Ponto) e o raio, valor real

Métodos:

|  |  |
| --- | --- |
| ***Construtor (\_\_init\_\_)*** | este método recebe os valores e cria um objeto . Default: centro 0,0 e raio 1 |
| ***Apresentação (\_\_str\_\_ e \_\_repr\_\_)*** | retorna uma string com os valores dos atributos |
|  |  |
| **==** | recebe como parâmetro um outro objeto da classe Círculo e retorna True se tiver os mesmos valores de atributo e False caso contrário. |
| **!=** | recebe como parâmetro um outro objeto da classe Círculo e retorna False se tiver os mesmos valores ou True caso contrário. |
| **>** | recebe como parâmetro um outro objeto da classe Círculo e retorna True se tiver maior raio e False caso contrário |
| **<** | recebe como parâmetro um outro objeto da Classe Círculo e retorna True se for tiver menor raio e False caso contrário |
|  |  |
| **+** | recebe como parâmetro um outro objeto da classe Círculo e retorna um novo Círculo cujo centro é o ponto mediano entre os círculos e o raio é a soma dos raios |
| **-** | recebe como parâmetro um outro objeto da classe Círculo e retorna um novo Círculo cujo centro é o ponto mediano entre os círculos e o raio é a subtração dos raios. |
| **\*** | recebe como parâmetro um número e retorna um novo círculo com mesmo centro e raio multiplicado pelo valor |
| **/** | recebe como parâmetro um número e retorna um novo círculo com mesmo centro e raio dividido pelo valor |
|  |  |
| Tipo | Recebe outro círculo e retorna o tipo ou **‘Não encostam’**  **Tipos:**  Circunferências tangentes externas: http://mundoeducacao.bol.uol.com.br/upload/conteudo/acir3.jpg  dOC= r1+ r2    http://mundoeducacao.bol.uol.com.br/upload/conteudo/acir2.jpgCircunferências tangentes Internas:  dOC= r1- r2    http://mundoeducacao.bol.uol.com.br/upload/conteudo/acir5(1).jpg  Circunferências concêntricas :  dCO= 0  a distância entre os centros é nula  http://mundoeducacao.bol.uol.com.br/upload/conteudo/acir4.jpg  dCO< r1+ r2  Circunferências secantes : |
| **Interior** | Recebe um ponto e retorna True se o ponto está dentro do círculo ou False, cc |

A classe Ponto está disponibilizada no arquivo classePonto.py e possui os seguintes métodos:

|  |  |
| --- | --- |
| ***Construtor (\_\_init\_\_)*** | este método recebe os valores e cria um objeto . Default: pto 0,0 |
| ***Apresentação (\_\_str\_\_ e \_\_repr\_\_)*** | retorna uma string com os valores dos atributos |
| **pontoMedio** | recebe obj pto e retorna um ponto com as coordenadas do ponto médio entre eles . |
| **alteraCoordenadas** | Recebe uma das coordenadas ou ambas e alterando as coordenadas deste ponto |
| **getX** | Retorna o valor da coordenada x |
| **getY** | Retorna o valor da coordenada y |
| **distAOrigem** | Recebe a distância deste ponto até a origem |
| **disEntre2Pontos** | Recebe um Ponto e retorna a distância entre elesade ou False , cc |
| **coefAngular** | Recebe um Ponto e retorna o coeficiente angular entre os pontos |
| **clonar** | Constrói outro Ponto com os mesmos atributos |
| **+** | Constrói novo ponto cujo x, y é a soma das coordenadas |
| **==** | Retorna True se os pontos têm mesmas coordenadas |
| **!=** | Retorna True se os pontos não têm mesmas coordenadas |
| **<** | Retorna True se 1º ponto está mais próximo da origem |
| **>** | Retorna True se 1º ponto está mais distante da origem |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A classe Data está disponibilizada no arquivo classeDataCompleta e possui os seguintes métodos:

|  |  |
| --- | --- |
| ***Construtor (\_\_init\_\_)*** | este método recebe os valores e cria um objeto . Default: : data atual |
| ***Apresentação (\_\_str\_\_ e \_\_repr\_\_)*** | retorna uma string com os valores dos atributos |
|  |  |
| **==** | recebe como parâmetro um outro objeto da classe Data e retorna True se tiver os mesmos valores de atributo e False caso contrário. |
| **!=** | recebe como parâmetro um outro objeto da classeData e retorna False se tiver os mesmos valores ou True caso contrário. |
| **>** | recebe como parâmetro um outro objeto da classe Data e retorna True se for mais recente que a outra e False caso contrário |
| **<** | recebe como parâmetro um outro objeto da Classe Data e retorna True se for mais antiga raio e False caso contrário |
|  |  |
| **-** | recebe como parâmetro um outro objeto da classe Data e retorna a quantidade de dias entre elas |
|  |  |
| **+** | Recebe um número de dias e retorna a data após/antes este intervalo de dias |
| **getMes** | Retorna o mês |
| **getDia** | | Retorna o dia |
| **getAno** | | Retorna o ano |
| **getMesExtenso** | | Retorna uma string com o nome do mês |
| **setMes** | | Recebe um valor e atualiza o atributo mês |
| **setDia** | | Recebe um valor e atualiza o atributo dia |
| **setAno** | | Recebe um valor e atualiza o atributo ano |

1. Sabendo que existe a classe Data (no arquivo classeDataCompleta.py) construa a classe Produto com:

* **Atributos**:

Identificação 🡪 string - default : ‘Produto’

CódigoBarras🡪 string - default: ''

Data de Fabricação 🡪 Data - default: data atual

Preço🡪 real, default: 0

* **Métodos desejados:**
  + - * Exibir
      * Idade : retorna o número de dias entre a data atual e a data de fabricação
      * ehSemelhante: recebe outro Produto e retorna True se tem mesma identificação
      * reajuste: recebe um valor percentual(x) e atualiza o preço em x%
      * get para cada atributo
      * set para cada atributo
      * Saber se dois produtos são iguais: mesmos atributos
      * Saber se um produto “é menor” que outro: se foi fabricado há menos tempo que o outro (menor idade). Caso os dois produtos tenham a mesma idade, considerar o preço

Ex P1 🡪 fabricado em 12/12/2019 custa 9000,00

P2 🡪 fabricado em 1/12/2019 custa 1000,00

P3 🡪 fabricado em 1/12/2019 custa 50,00

P1 < P2 🡪 True

P2<P3 🡪 False