[**ExerciciosClasses**](https://wiki.python.org.br/ExerciciosClasses?action=fullsearch&context=180&value=linkto%3A%22ExerciciosClasses%22)

Voltar para a [ListaDeExercicios](https://wiki.python.org.br/ListaDeExercicios)

1. **~~Classe Bola:~~**~~Crie uma classe que modele uma bola:~~
   1. ~~Atributos: Cor, circunferência, material~~
   2. ~~Métodos: trocaCor e mostraCor~~
2. **~~Classe Quadrado:~~**~~Crie uma classe que modele um quadrado:~~
   1. ~~Atributos: Tamanho do lado~~
   2. ~~Métodos: Mudar valor do Lado, Retornar valor do Lado e calcular Área;~~
3. **~~Classe Pessoa:~~**~~Crie uma classe que modele uma pessoa:~~
   1. ~~Atributos: nome, idade, peso e altura~~
   2. ~~Métodos: Envelhercer, engordar, emagrecer, crescer. Obs: Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.~~
4. **~~Classe Conta Corrente:~~**~~Crie uma classe para implementar uma conta corrente. A classe deve possuir os seguintes atributos: número da conta, nome do correntista e saldo. Os métodos são os seguintes: alterarNome, depósito e saque; No construtor, saldo é opcional, com valor default zero e os demais atributos são obrigatórios.~~
5. **~~Classe TV:~~**~~Faça um programa que simule um televisor criando-o como um objeto. O usuário deve ser capaz de informar o número do canal e aumentar ou diminuir o volume. Certifique-se de que o número do canal e o nível do volume permanecem dentro de faixas válidas~~.
6. **~~Classe Bichinho Virtual:~~**~~Crie uma classe que modele um Tamagushi (Bichinho Eletrônico):~~
   1. ~~Atributos: Nome, Fome, Saúde e Idade b. Métodos: Alterar Nome, Fome, Saúde e Idade; Retornar Nome, Fome, Saúde e Idade Obs: Existe mais uma informação que devemos levar em consideração, o Humor do nosso tamagushi, este humor é uma combinação entre os atributos Fome e Saúde, ou seja, um campo calculado, então não devemos criar um atributo para armazenar esta informação por que ela pode ser calculada a qualquer momento.~~
7. **Classe Ponto e Retangulo:** Faça um programa completo utilizando funções e classes que:
   1. Possua uma classe chamada Ponto, com os atributos x e y.
   2. Possua uma classe chamada Retangulo, com os atributos largura e altura.
   3. Possua uma função para imprimir os valores da classe Ponto
   4. Possua uma função para encontrar o centro de um Retângulo.
   5. Você deve criar alguns objetos da classe Retangulo.
   6. Cada objeto deve ter um vértice de partida, por exemplo, o vértice inferior esquerdo do retângulo, que deve ser um objeto da classe Ponto.
   7. A função para encontrar o centro do retângulo deve retornar o valor para um objeto do tipo ponto que indique os valores de x e y para o centro do objeto.
   8. O valor do centro do objeto deve ser mostrado na tela
   9. Crie um menu para alterar os valores do retângulo e imprimir o centro deste retângulo.
8. **~~Classe Bomba de Combustível:~~**~~Faça um programa completo utilizando classes e métodos que:~~
   1. ~~Possua uma classe chamada bombaCombustível, com no mínimo esses atributos:~~
      1. ~~tipoCombustivel.~~
      2. ~~valorLitro~~
      3. ~~quantidadeCombustivel~~
   2. ~~Possua no mínimo esses métodos:~~
      1. ~~abastecerPorValor( ) – método onde é informado o valor a ser abastecido e mostra a quantidade de litros que foi colocada no veículo~~
      2. ~~abastecerPorLitro( ) – método onde é informado a quantidade em litros de combustível e mostra o valor a ser pago pelo cliente.~~
      3. ~~alterarValor( ) – altera o valor do litro do combustível.~~
      4. ~~alterarCombustivel( ) – altera o tipo do combustível.~~
      5. ~~alterarQuantidadeCombustivel( ) – altera a quantidade de combustível restante na bomba.~~

~~OBS: Sempre que acontecer um abastecimento é necessário atualizar a quantidade de combustível total na bomba.~~

1. **~~Classe carro:~~**~~Implemente uma classe chamada Carro com as seguintes propriedades:~~
   1. ~~Um veículo tem um certo consumo de combustível (medidos em km / litro) e uma certa quantidade de combustível no tanque.~~
   2. ~~O consumo é especificado no construtor e o nível de combustível inicial é 0.~~
   3. ~~Forneça um método andar( ) que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.~~
   4. ~~Forneça um método obterGasolina( ), que retorna o nível atual de combustível.~~
   5. ~~Forneça um método adicionarGasolina( ), para abastecer o tanque. Exemplo de uso:~~
   6. ~~meuFusca = Carro(15); # 15 quilômetros por litro de combustível.~~
   7. ~~meuFusca.adicionarGasolina(20); # abastece com 20 litros de combustível.~~
   8. ~~meuFusca.andar(100); # anda 100 quilômetros.~~

~~meuFusca.obterGasolina() # Imprime o combustível que resta no tanque.~~

1. **~~Classe Conta de Investimento:~~**~~Faça uma classe contaInvestimento que seja semelhante a classe contaBancaria, com a diferença de que se adicione um atributo taxaJuros. Forneça um construtor que configure tanto o saldo inicial como a taxa de juros. Forneça um método adicioneJuros (sem parâmetro explícito) que adicione juros à conta. Escreva um programa que construa uma poupança com um saldo inicial de R$1000,00 e uma taxa de juros de 10%. Depois aplique o método adicioneJuros() cinco vezes e imprime o saldo resultante.~~
2. **~~Classe Funcionário:~~**~~Implemente a classe Funcionário. Um empregado tem um nome (um string) e um salário(um double). Escreva um construtor com dois parâmetros (nome e salário) e métodos para devolver nome e salário. Escreva um pequeno programa que teste sua classe.~~
3. ~~Aprimore a classe do exercício anterior para adicionar o método aumentarSalario (porcentualDeAumento) que aumente o salário do funcionário em uma certa porcentagem.~~
   1. ~~Exemplo de uso:~~
   2. ~~harry=funcionário("Harry",25000)~~

~~harry.aumentarSalario(10)~~