

## ExerciciosClasses

---

Voltar para a [ListaDeExercicios](#)

---

- Classe Bola:** Crie uma classe que modele uma bola:
  - Atributos: Cor, circunferência, material
  - Métodos: trocaCor e mostraCor
- Classe Quadrado:** Crie uma classe que modele um quadrado:
  - Atributos: Tamanho do lado
  - Métodos: Mudar valor do Lado, Retornar valor do Lado e calcular Área;
- Classe Retangulo:** Crie uma classe que modele um retangulo:
  - Atributos: LadoA, LadoB (ou Comprimento e Largura, ou Base e Altura, a escolher)
  - Métodos: Mudar valor dos lados, Retornar valor dos lados, calcular Área e calcular Perímetro;
  - Crie um programa que utilize esta classe. Ele deve pedir ao usuário que informe as medidas de um local. Depois, deve criar um objeto com as medidas e calcular a quantidade de pisos e de rodapés necessárias para o local.
- Classe Pessoa:** Crie uma classe que modele uma pessoa:
  - Atributos: nome, idade, peso e altura
  - 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.
- 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.
- 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.
- Classe Bichinho Virtual:** Crie uma classe que modele um Tamagushi (Bichinho Eletrônico):
  - Atributos: Nome, Fome, Saúde e Idade
  - 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.
- Classe Macaco:** Desenvolva uma classe Macaco, que possua os atributos nome e bucho (estomago) e pelo menos os métodos comer(), verBucho() e digerir(). Faça um programa ou teste interativamente, criando pelo menos dois macacos, alimentando-os com pelo menos 3 alimentos diferentes e verificando o conteúdo do estomago a cada refeição. Experimente fazer com que um macaco coma o outro. É possível criar um macaco canibal?
- Classe Ponto e Retangulo:** Faça um programa completo utilizando funções e classes que:
  - Possua uma classe chamada Ponto, com os atributos x e y.
  - Possua uma classe chamada Retangulo, com os atributos largura e altura.
  - Possua uma função para imprimir os valores da classe Ponto
  - Possua uma função para encontrar o centro de um Retângulo.
  - Você deve criar alguns objetos da classe Retangulo.
  - Cada objeto deve ter um vértice de partida, por exemplo, o vértice inferior esquerdo do retângulo, que deve ser um objeto da classe Ponto.
  - 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.
  - O valor do centro do objeto deve ser mostrado na tela
  - Crie um menu para alterar os valores do retângulo e imprimir o centro deste retângulo.
- Classe Bomba de Combustível:** Faça um programa completo utilizando classes e métodos que:
  - Possua uma classe chamada bombaCombustível, com no mínimo esses atributos:
    - tipoCombustivel.
    - valorLitro
    - quantidadeCombustivel
  - Possua no mínimo esses métodos:
    - abastecerPorValor( ) – método onde é informado o valor a ser abastecido e mostra a quantidade de litros que foi colocada no veículo
    - abastecerPorLitro( ) – método onde é informado a quantidade em litros de combustível e mostra o valor a ser pago pelo cliente.

- iii. `alterarValor()` – altera o valor do litro do combustível.
  - iv. `alterarCombustivel()` – altera o tipo do combustível.
  - v. `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.

11. **Classe carro:** Implemente uma classe chamada Carro com as seguintes propriedades:

- a. Um veículo tem um certo consumo de combustível (medidos em km / litro) e uma certa quantidade de combustível no tanque.
- b. O consumo é especificado no construtor e o nível de combustível inicial é 0.
- c. 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.
- d. Forneça um método `obterGasolina()`, que retorna o nível atual de combustível.
- e. Forneça um método `adicionarGasolina()`, para abastecer o tanque. Exemplo de uso:

```
meuFusca = Carro(15);           # 15 quilômetros por litro de combustível.  
meuFusca.adicionarGasolina(20); # abastece com 20 litros de combustível.  
meuFusca.andar(100);           # anda 100 quilômetros.  
meuFusca.obterGasolina()       # Imprime o combustível que resta no tanque.
```

12. **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 `adiconeJuros` (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 `adiconeJuros()` cinco vezes e imprime o saldo resultante.
13. **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.
14. Aprimore a classe do exercício anterior para adicionar o método `aumentarSalario` (porcentualDeAumento) que aumente o salário do funcionário em uma certa porcentagem.
- Exemplo de uso:

```
harry=funcionário("Harry",25000)  
harry.aumentarSalario(10)
```

15. **Classe Bichinho Virtual++:** Melhore o programa do bichinho virtual, permitindo que o usuário especifique quanto de comida ele fornece ao bichinho e por quanto tempo ele brinca com o bichinho. Faça com que estes valores afetem quão rapidamente os níveis de fome e tédio caem.
16. Crie uma "porta escondida" no programa do bichinho virtual que mostre os valores exatos dos atributos do objeto. Consiga isto mostrando o objeto quando uma opção secreta, não listada no menu, for informada na escolha do usuário. Dica: acrescente um método especial `str()` à classe `Bichinho`.
17. Crie uma Fazenda de Bichinhos instanciando vários objetos bichinho e mantendo o controle deles através de uma lista. Imite o funcionamento do programa básico, mas ao invés de exigir que o usuário tome conta de um único bichinho, exija que ele tome conta da fazenda inteira. Cada opção do menu deveria permitir que o usuário executasse uma ação para todos os bichinhos (alimentar todos os bichinhos, brincar com todos os bichinhos, ou ouvir a todos os bichinhos). Para tornar o programa mais interessante, dê para cada bichinho um nível inicial aleatório de fome e tédio.

---

Voltar para a [ListaDeExercicios](#)

**Sobre esta página**

ExerciciosClasses (editada pela última vez em 2008-09-26 14:07:17 por localhost)

[Visualizar Texto](#) | [Visualizar Impressão](#) | [Information](#) | [Fazer Usuário Acompanhar](#) | [Anexos](#)



"Python" e os logos de Python são marcas registradas da [Python Software Foundation](#), usadas aqui mediante permissão da mesma. O conteúdo deste site está disponível sob os termos da [Creative Commons Attribution 2.5](#) exceto quando explicitamente especificado outra licença.