



Atividades práticas

Em nosso capítulo, percebemos a importância em entender os diferentes tipos de paradigmas de programação, com o foco em Orientação ao Objeto. Aprendemos também sobre os pilares da orientação ao objeto: Herança, Encapsulamento, Abstração e Polimorfismo, bem como suas utilidades quando falamos em usabilidade de código, eficiência, fácil manutenção do código, escalabilidade, reutilização dos códigos e outros aspectos que facilitam a vida do programador. Vamos fazer práticas que colaboram com todo o nosso aprendizado destinado a esse capítulo.

- ✓ **1ª tarefa: Floresta** - Crie uma classe animal que receba quatro atributos e quatro métodos. Na main, crie oito objetos que imprima seus atributos e métodos.
 - Você pode definir os atributos e métodos a serem impressos, utilize sua imaginação para realizar essa tarefa.

- ✓ **2ª tarefa: Carros e seus modelos** - Crie uma classe chamada carros com três modelos de carros, junto aos atributos (Dados como parâmetros): consumo de gasolina, cor, tipo do motor, marca e odômetro e seu método consumo().
 - Se o carro fizer o consumo de 15km a cada 1 litro de gasolina, imprima o modelo do carro (escolhido por você) é econômico.

- Se o carro fizer o consumo abaixo de 15km a cada 1 litro de gasolina, imprima o modelo do carro (escolhido por você) é de um consumo padrão alto.
- Crie, no mínimo, três objetos carros de modelos distintos e imprima-os.
- Informe qual o melhor carro seguindo o custo-benefício Km/L.

✓ **3ª tarefa: Banco Simples** - Crie uma classe chamada ContaBancaria que tenha os atributos privados saldo e titular(nome da pessoa titular da conta). Implemente os métodos públicos get_saldo(), depositar(valor) e sacar(valor) para acessar e modificar o saldo da conta de forma controlada. O método sacar(valor) deve verificar se há saldo suficiente antes de efetuar o saque.

- Caso o saque seja maior que o saldo, imprima uma mensagem informando que o saque não pode ser efetuado.
- Caso o saque seja exatamente igual ao saldo, imprima uma mensagem informando que a conta está zerada.
- Importante enfatizar que o titular pode realizar depósitos e saques, porém não pode mexer no saldo da conta de forma livre, por isso essa deve ser privada a classe.
- Os depósitos só podem ser feitos com limite de 10.000 reais, sendo maior que esse valor o sistema deve retornar uma mensagem pedindo para ele fazer um novo depósito com esse limite.
- Os saques têm um limite de 50.000 por saque, caso o titular peça um valor maior que esse, deverá retornar uma mensagem informando que o limite de saque não é permitido.

- ✓ **4ª tarefa: Loja Online** - Crie uma classe chamada Produto que possui os atributos privados nome, preco e qtd_estoque. Implemente os métodos públicos get_nome(), get_preco(), get_estoque(), atualizar_preco(novo_preco) e atualizar_estoque(quantidade) para acessar e modificar informações sobre produtos. O método atualizar_estoque(quantidade) deve verificar se a quantidade atualizada não deixará o estoque negativo.

- Se o produto estiver com cinco unidades ou menos, imprima a mensagem (Produto em quantidade baixa, fazer novo pedido com urgência).
- Se o produto estiver com mais de 100 unidades, diminua o preço do produto em 40% e imprima a mensagem (Quantidade alta do \$produto, favor oferecer com 40% de desconto).
- Retorne todas as informações do produto, seguindo os métodos e atributos declarados.

- ✓ **5ª tarefa: Secretaria Escolar** - Crie uma classe chamada Aluno com os atributos privados nome, matricula e notas (uma lista de quatro notas). Implemente os métodos públicos get_nome(), get_matricula(), get_media() e adicionar_nota(nota) para acessar e modificar as informações dos alunos. O método get_media() deve calcular a média das notas do aluno.

- Devemos realizar a leitura de dados do nome do aluno, da sua matrícula e das quatro notas.
- O programa deverá retornar o nome do aluno, a matrícula fornecida e a média desse aluno.

- Se na leitura de dados as notas forem maiores que 10 ou menores que 0, utilizar tratamento de exceções retornando (Erro ao digitar as notas).
 - Caso a média seja maior ou igual a 7,0 imprima (Aluno aprovado, parabéns!). Se a média for menor que 7,0 e maior ou igual a 4,0 imprima (Aluno está em processo de recuperação). E se a média for menor que 4,0 imprima (O Aluno está reprovado).
- ✓ **6ª tarefa: Formas Geométricas** - Crie uma classe abstrata chamada Forma com um método abstrato `calcular_area()`. Crie três classes derivadas: Quadrado, Círculo e Triângulo. Implemente o método `calcular_area()` em cada classe de forma apropriada para cada forma geométrica. Crie objetos dessas classes e calcule suas áreas.
- Importante enfatizar que não pode criar o objeto na main vindo de formas, mas sim nas formas que herdam.
 - Não poderá ser digitado números negativos, fazer tratamento de exceção que imprima (entradas inválidas).
 - A impressão do valor da área deverá do tipo Double. Os lados do triângulo poderá ser do tipo inteiro ou float.
 - Para o cálculo da área: Quadrado recebe somente o valor do lado, círculo recebe o valor do raio e triângulo recebe o valor dos 3 lados do triângulo.
 - Se atente nos critérios de existência do triângulo (Só irá existir um triângulo se, somente se, os seus lados obedeceram à seguinte regra: um de seus lados deve ser maior que o valor da diferença dos outros dois lados e menor que a soma dos outros dois lados.), pois nem sempre poderá ser formado um

triângulo com base nos lados. Não sendo possível imprima (Os lados informados não compõem um triângulo).

- Para calcular a área do triângulo, considere as fórmulas para ele caso seja equilátero ou não. Área do triângulo não equilátero: $\frac{b \cdot h}{2}$ e do triângulo equilátero: $\frac{l^2 \sqrt{3}}{4}$. Sendo que você poderá (opcionalmente) fazer a sobrecarga de métodos para receber a base e a altura para uma fórmula e somente o lado na outra forma, respectivamente. (Para o cálculo de raiz 3, considere 1,7 para efeitos de cálculo).
- O cálculo da altura deverá obedecer a fórmula de Pitágoras, você passará somente os lados e o programa deve calcular a altura, caso o triângulo não seja equilátero. Sendo posteriormente passado na instância junto ao valor da base.
- O valor da base deve ser o lado diferente dos demais lados caso o triângulo seja isóscele (dois lados iguais). E caso o triângulo seja escaleno (3 lados diferentes) o valor da base não deverá ser o maior valor, pois facilitará nossos cálculos e a lógica do programa.

✓ **7ª tarefa: Meios de Transporte Polimórficos** - Crie uma classe base chamada MeioDeTransporte com um método mover(). Crie classes derivadas para diferentes meios de transporte, como Carro, Avião, Moto, Jato, Quadriciclo e Bicicleta. Implemente o método mover() de maneira diferente para cada meio de transporte, refletindo como eles se deslocam.

- Realize entradas de usuário para realizar a tarefa acima.

- Importante nessa tarefa é que você faça uma função que realize o método `mover()` para todos os transportes supracitados.
- ✓ **8ª tarefa: Sistema de Funcionários** - Crie uma classe chamada `Funcionario` com os atributos `nome`, `salario`, `matrícula` e `cargo`. Implemente os métodos públicos `get_nome()`, `get_salario()`, `get_cargo()`, e os métodos do tipo privado `aumentar_salario(percentual)` e `promover(novo_cargo)` para acessar e atualizar as informações dos funcionários.
 - O método `aumentar_salario(percentual)` deve calcular e aplicar um aumento salarial baseado no `percentual` fornecido do tipo `float`.
 - Nessa tarefa, o encapsulamento é utilizado para controlar o acesso aos atributos da classe e garantir que as operações sejam realizadas de maneira segura e controlada através dos métodos públicos e privados. Isso permite que você mantenha a coesão no encapsulamento de dados, evitando acesso direto e não autorizado aos atributos e/ou métodos da classe.
 - Realize a impressão dos dados do usuário passados nos parâmetros. Logo depois, crie uma pergunta, na entrada de dados, se esse funcionário foi promovido. Caso sim, informe a nova matrícula, através da entrada de dados, e reimprima todas as informações do funcionário de forma atualizada.
 - Para acessar e /ou modificar a matrícula, crie dois métodos (`Get` e `Set`). O método `Get` e `Set` devem ser públicos para poder realizar o acesso na `Main`.
 - Para acessar e /ou modificar o cargo crie dois métodos (`Get` e `Set`). O método `Get` e `Set` devem ser públicos para poder realizar o acesso na `Main`.

- Para acessar e /ou modificar o salário, crie dois métodos (Get e Set).
O método Get e Set devem ser públicos para poder realizar o acesso na Main.