

## Exercícios sobre Classes Abstratas.

Modele utilizando a ferramenta Umbrello e desenvolva a aplicação utilizando a linguagem C++.

1. Escreva uma superclasse abstrata encapsulando um veículo. Um veículo tem dois atributos: o nome de seu dono e seu número de rodas, além do atributo ano de fabricação, e valor que o proprietário pagou na aquisição do veículo. Além do método abstrato `valor_de_venda`. Esta classe possui duas classes não abstratas uma encapsulando uma bicicleta, e outra encapsulando um veículo motorizado. Uma bicicleta sofre uma desvalorização de 15% por ano. Um veículo motorizado contém os seguintes atributos adicionais: um atributo cilindrada, em litros, e um método que compute e retorne a média em cavalo-vapor – o número de litros vezes o número de rodas. Um veículo motorizado sofre uma desvalorização de 8% por ano.
2. Escreva uma superclasse abstrata encapsulando alguma comida. Ela contém os seguintes atributos: descrição, quantidade calórica. Também o método abstrato que tenha o número de porções como parâmetro e retorne o número de calorias. Esta classe tem duas subclasses não abstratas: uma encapsulando uma comida líquida (como uma bebida, suco, açaí. Por exemplo) e outra encapsulando uma fruta. Uma comida líquida tem um atributo adicional: sua viscosidade. Uma fruta tem um atributo adicional: validade.
3. Escreva uma superclasse abstrata encapsulando um vestibulando. Um vestibulando tem dois atributos: o nome do candidato e a faculdade que o candidato quer cursar. Esta classe tem duas subclasses não abstratas: uma encapsulando um candidato ainda não graduado e a outra encapsulando um candidato já graduado. O primeiro candidato tem dois atributos adicionais: disciplinas cursadas, tempo do curso. O segundo candidato tem um atributo adicional: faculdade de origem. Também tem um método que retornará “de dentro” se a faculdade de origem for a mesma que a faculdade pretendida. Caso contrário “de fora”.
4. Escreva uma superclasse abstrata encapsulando férias. Férias tem dois atributos: um orçamento e destino. Tem um método abstrato que retorna em quanto as férias ficam abaixo ou acima do orçamento. Esta classe tem duas subclasses não abstratas: uma encapsulando férias com tudo incluído e a outra encapsulando férias compradas parcialmente. Férias com tudo incluído tem três atributos adicionais: uma marca, uma classificação, expressa por número de estrelas e um preço. Férias parciais tem dois atributos adicionais: um conjunto de itens (hotel, refeições, passagens) e um conjunto de custos correspondentes.
5. Crie uma hierarquia de classes para representar os diferentes tipos de funcionários de um escritório que tem os seguintes cargos: gerente, assistente, vendedor. Escreva uma classe base abstrata chamada `Funcionario` que declara um método abstrato:

`double calcula_salario()`

Esta classe também deve definir os seguintes atributos: `nome` (tipo `string`), `matricula` (tipo `string`) e `salario_base` (tipo `double`). Esta classe abstrata deverá ser estendida pelas outras classes representativas dos tipos de funcionários, portanto devem ser escritas as classes `Gerente`, `Assistente` e `Vendedor`. Em cada classe deve-se sobrescrever o método `calcula_salario` de forma que o cálculo do salário é feito assim: O gerente recebe duas vezes o `salario_base`, o assistente recebe o `salario_base` e o vendedor recebe o `salario_base` mais uma comissão definida no construtor de sua classe. Crie uma aplicação que possibilite o cadastro dos funcionários, e calcule a folha salarial dos funcionários e imprime o valor total que esta folha dará para a empresa.

6. Escreva uma classe abstrata chamada CartaoWeb. Essa classe representa todos os tipos de cartões web e conterá apenas um atributo: destinatario (tipo string). Nessa classe você deverá também declarar o método abstrato show\_mensagem. Crie classes filhas da classe CartaoWeb: DiaDosNamorados, Natal, Aniversario. Cada uma dessas classes deve conter um método construtor que receba o nome do destinatário do cartão. Cada classe também deve implementar o método show\_message(), mostrando uma mensagem ao usuário com seu nome e que seja específica para a data de comemorativa do cartão.
7. Implemente uma classe abstracta de nome Forma onde são declarados dois métodos abstractos:
- float calcularArea();
  - float calcularPerimetro();
- Crie, como subclasse de Forma, uma classe de nome Rectangulo cujas instâncias são caracterizadas pelos atributos lado e altura ambos do tipo float. Implemente na classe Rectangulo os métodos herdados de Forma e outros que ache necessários.
- Crie, como subclasse de Forma, uma classe de nome Circulo cujas instâncias são caracterizadas pelo atributo raio do tipo float. Implemente na classe Circulo os métodos herdados de Forma e outros que ache necessários. Nota: poderá aceder ao valor de Pi fazendo Math.Pi.
- Crie, como subclasse de Rectangulo, uma classe de nome Quadrado cujas instâncias são caracterizadas por terem os atributos lado e altura com o mesmo valor.
- Elabore um programa de onde é declarado um vector que armazene Forma. Nesse vector devem ser guardadas instâncias de Rectangulo, Circulo e Quadrado seguindo uma ordem aleatória.