

Polimorfismo

1. O mundo das formas é muito rico. Crie uma interface (em C++ uma classe abstract apenas com métodos virtuais = 0) chamada de **Forma**. A interface mais abstrata, **Forma**, deve possuir dois métodos: `std::string get_cor()` e `std::string get_nome()`. A mesma tem duas sub-interfaces: **Forma2D** e **Forma3D**. A primeira, deve conter os métodos `double get_area()` e `double get_perimetro()`. A segunda tem um único método `double get_volume()`. Implemente as classes: **Quadrado**, **Circulo**, **Triangulo**, **Esfera** e **Cubo**.
2. Uma farmácia necessita controlar todos os produtos que comercializa. Sabe-se que nesse estabelecimento os produtos comercializados são medicamentos e artigos de higiene. Os medicamentos possuem código, descrição, preço de compra, percentual de lucro e valor do desconto do laboratório. Os artigos de higiene possuem código, descrição, preço de compra e percentual de lucro. Sabe-se que um artigo de higiene gera como lucro 30% do preço de compra.
 - Crie as classes que representem o contexto descrito;
 - Crie na classe **Remedio** o seguinte método:
 - **gerarPrecoDeVenda**: método sem parâmetros que deverá retornar o preço de venda do remédio. O preço de venda é a soma do preço de compra mais o percentual de lucro do preço de compra, menos o desconto do laboratório.
 - Crie na classe **ArtigoHigiene** o seguinte método:
 - **gerarPrecoDeVenda**: método sem parâmetros que deverá retornar o preço de venda do artigo de higiene. O preço de venda é a soma do preço de compra mais o percentual de lucro do preço de compra.
 - Crie um **main.cpp**. Crie dois remédios e dois artigos de higiene. Utilize todos os métodos das classes criadas. Imprima todos os dados dos produtos. Importante! Para resolução deste exercício você deve empregar, obrigatoriamente, os conceitos de herança e polimorfismo.

Considerações!

- Todos os exercícios devem conter **.h**, **.cpp**, e **main.cpp**;
- O seu **main.cpp** deve conter, minimamente, instruções para criação (instanciação de objetos) e chamadas das funções implementadas.