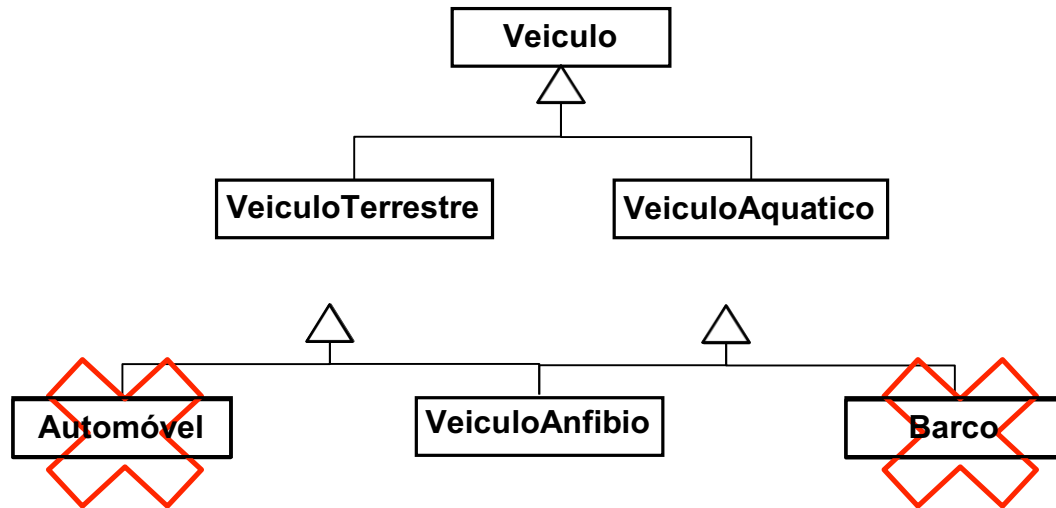


Lista de Exercícios – III

Herança

1. Escreva um programa que defina uma classe Poligono com um construtor que recebe valor para Comprimento e Altura. Defina duas subclasses Triangulo e Retangulo, que calculam a área por uma função `double area()`. No main, defina duas variáveis Triangulo e Retangulo e então chame a função `area()` desses objetos.
2. Crie uma classe *Empresa* capaz de armazenar os dados de uma empresa (Nome, Endereço, Cidade, Estado, CEP e Telefone). Inclua um construtor sem argumentos e um com argumentos para inicialização dos atributos. Crie métodos que funcionem como *getter* e *setter* e *print*.
 - Utilize a classe *Empresa* como base para criar a classe *Restaurante*. Esta classe derivada deve conter atributos que representem o tipo de comida e o preço médio de um prato. Crie um construtor para esta classe que chame explicitamente o construtor da classe *Empresa*, um *getter* e um *setter*, além de um método *print*, que utiliza o método *print* da classe base. Crie um *driver(programa principal - main)* para testar sua aplicação.
3. Crie uma classe *Pessoa* e a classe *Veiculo*. Na classe *Pessoa*, deve conter nome e `rg(int)`. A classe *Veiculo* possui o peso, a velocidade máxima, preço e dono(instancia de *Pessoa*). Inclua um construtor sem argumentos e um com argumentos para inicialização dos atributos tanto para classe *Pessoa* como para classe *Veiculo*. Crie métodos que funcionem como *getter* e *setter* e *toString*.
 - Crie a classe *Motor*, contendo o número de cilindros e a potência. Inclua um construtor sem argumentos e um com argumentos para inicialização dos atributos. Crie métodos que funcionem como *getter* e *setter* e *toString*.
 - Crie a classe *CarroPasseio*, derivada das classes *Veiculo* e *Motor*. Inclua atributos como modelo e cor. Crie um construtor para esta classe que chame explicitamente o construtor das classes base e preencha suas informações
 - Crie a classe *Caminhao*, também derivada das classes *Motor* e *Veiculo*. Inclua os atributos toneladas, altura máxima e comprimento. Crie um construtor para esta classe que chame explicitamente o construtor das classes base e um método *toString* que redefino e método da classe base e que também utiliza o método *toString* da classe base na sua implementação, além de adicionar suas próprias informações.
 - Crie um *driver (programa principal - main)* para testar sua aplicação. Implemente os métodos getters e setters necessários para testar seu programa.
4. Implemente a hierarquia de herança múltipla definida pelo diagrama UML abaixo. Em cada classe, defina um construtor e defina dois atributos, com *getter*, *setter* e *toString*. Implemente classe *VeiculoAnfibio* como classe derivada e crie um método *toString*, que imprime todos os dados de um veículo anfíbio. As classes marcadas com x não precisam ser implementadas.



Referências

- Programação C/C++. Prof. Márcio Sarroglia Pinho.
<https://www.inf.pucrs.br/~pinho/PRGSWB/OO/oocpp.html>.
- <https://www.inf.pucrs.br/~pinho/CPP/Exercicios/Classes/ExercicioVeiculo.html>
- <https://www.inf.pucrs.br/~pinho/CPP/Exercicios/Classes/ExercicioAgenda.html>