

EXAME DE SUBSTITUIÇÃO DE 1ª PARCIAL - DAINF/UTFPR Curitiba-Central – Tec Prog CSE20
Prova sobre linguagem C++ (versão 2003) / Orientação a Objetos / Diagrama de Classes em UMLNome do Aluno: _____ Turma: S71
Curso: _____ Horário de Começo: _____ Horário de Fim: _____**Leia toda a prova, pois os enunciados estão completados uns nos outros.****Utilize os bons princípios de projeto e programação orientada a objetos.****Em tempo, questões de mesmo peso, idem para os itens dentro delas (quando houver).**

(Questão - 1) (a) Elabore um diagrama de classes em *UML* das classes solicitadas, nesta e nas demais questões, com seus relacionamentos e podendo ter (parte de) seus atributos e métodos, antes de elaborar o código pertinente. (b) Em um programa C++ (*console*) para ‘simulação’ de trânsito, crie uma classe *Veiculo* que terá um atributo protegido *string* chamado *nome*, bem como atributos protegidos *float* chamados de *x* e *y* com seus respectivos *gets*. (c) Esta classe deve ter um método virtual puro chamado *atualizar*. Ainda, esta classe deve ter um método *mostrar* que chama *atualizar* e daí apresenta os valores de *nome*, *x* e *y*. (d) Essa classe permitirá duas classes derivadas dela, nomeadamente *Carro* (derivada diretamente) e *Viatura* (derivada indiretamente), sendo que essas três classes devem estar em um espaço de nomes (*namespace*) chamado *Automoveis*. Ainda, nessas classes requisitadas, assim como nas das demais questões, deve-se usar *const* sempre que possível.

(Questão - 2) (a) Crie uma classe *Carro* (derivada de *Veiculo*), com atributos protegidos inteiro *multas* e real *fator*. (b) O método *atualizar* será redefinido em *Carro* como virtual de maneira tal a permitir que se possa incrementar o *x* e *y* com *fator*. (c) Deve-se sobrecarregar o operador de recebimento com soma (*operator+=*) na classe *Carro* com um parâmetro *param*. (d) No método *atualizar* de *Carro*, deve-se chamar o operador de recebimento parametrizado com *fator*.

(Questão - 3) (a) Crie uma classe *Viatura* (derivada de *Carro*), com um atributo privado booleano *sirene*. (b) O método *atualizar* será redefinido em *Viatura* de maneira tal a permitir que se possa incrementar o *x* e *y* em $2 * \text{fator}$. (c) Deve-se sobrecarregar o operador de autossoma (*operator++*) na classe *Viatura* para incrementar *x* e *y* em $2 * \text{fator}$. (d) No método *atualizar* de *Viatura*, deve-se chamar o operador de autossoma, de forma tal a permitir o incremento em *x* e *y* como definido no item (b) justo acima.

(Questão - 4) (a) No contexto do programa C++ em questão, crie uma classe chamada *Frota*, usando o componente *set* (conjunto) da STL, para endereços de objetos que possam ser apontados como *Veiculo*. (b) Essa classe deve ter método de inclusão, lembrando que *set* não permite repetir elementos por definição. (c) Essa classe deve ter método de exclusão de todos os elementos, bem como deve ter um método de percorrimento que comece pelo último elemento e vá até primeiro, o qual chama o método *mostrar* de cada objeto apontado enquanto *Veiculo*. (d) Por fim, essa classe em questão, bem como a classe *Simulador* devem estar em um espaço de nomes chamado *Gerenciadores*. **Obs.:** Alternativamente ao componente *set* pode-se compor a classe *Frota* usando outro componente da STL, mas aí valendo no máximo 1/2 da questão.

(Questão - 5) (a) Crie uma classe *Simulador*, com um método *executar*, sendo que a instância ou o objeto dela, na função *main*, chamar-se-á *simulador*. Em *Simulador* deve haver como atributos dois objetos de *Carro* tendo os seguintes valores inicializados via construtora: para o 1º, como segue, *nome* = “cidadão”, *x* = 2, *y* = 2, *multas* = 0 e *fator* = 0,5; para o 2º, como segue, *nome* = “meliante”, *x* = 50, *y* = 50, *multas* = 500 e *fator* = 1. Em *Simulador*, haverá também como atributo uma instância de *Viatura*, tendo os seguintes valores inicializado via construtora *nome* = “policial”, *x* = 0, *y* = 0, *multas* = -1, *fator* = 3 e *sirene* = verdadeiro. (b) Em *simulador* ainda haverá como atributo privado um objeto de *Frota* no qual será inserido todos os endereços de objetos pertinentes usando *cast* (forjador) seguro apropriado. (c) Todos os objetos agregados em *Simulador* devem ser criados estaticamente. (d) No método *executar*, deve ter um laço de repetição que: (1) mostra a evolução de posição de todas as viaturas via objeto pertinente de *Frota* (em quadro de polimosfirmo). (2) para quando a ‘polícia’ alcançar (*i.e.*, *x* e *y* iguais) ou ultrapassar (*i.e.*, *x* e *y* maiores que o) o ‘meliante’ (o capturando) ou quando passar de cem iterações o laço (sem o capturar).