

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba Central
Departamento Acadêmico de Informática (DAINF). Disciplina: Técnicas de Programação – CSE20
Prova sobre linguagem C++ (2003) e afins – Exame de substituição de primeira parcial

Nome do(a) Discente: _____ Turma: S73
Curso: _____ Horário de Começo: _____ Horário de Fim: _____

Leia toda a prova, pois os enunciados estão completados uns nos outros. Ainda, utilize os bons princípios de programação orientada a objetos. Por fim, questões de mesmo peso, idem para os tópicos dentro delas.

(Questão - 1) (a) Primeiramente, antes das demais questões abaixo, elabore um diagrama em *UML* das classes solicitadas nelas e de seus relacionamentos. (b) Nessas classes abaixo requisitadas, deve-se usar *const* quando apropriado na modelagem e implementação. (c) Explique sucintamente se há caso de polimorfismo previsto nas questões e onde (se for o caso), salientando se o fato da classe veículo ser abstrata teria papel nisto. (d) cite três padrões de projeto (sendo um deles obrigatoriamente o apresentado por si em classe) e aplique um dos padrões de projeto citados a alguma parte do código solicitado nas questões que seguem.

(Questão - 2) (a) Em um programa C++ (*console*) para simulação de trânsito, crie uma classe *Veiculo* que terá um atributo protegido *char**, chamado *nome*, o qual deve ser alocado e desalocado dinamicamente. (b) Esta classe também terá mais dois atributos, sendo eles protegidos *float* chamados de *x* e *y*, cada qual com seu respectivo *get*. (c) Ainda, essa classe deve ter um método virtual puro chamado *atualizar*. (d) Por fim, aquela classe deve ter um método *mostrar* que chama *atualizar* e daí mostra os valores de *nome*, *x* e *y*.

(Questão - 3) (a) A classe *Veiculo* permitirá duas classes derivadas dela: *Moto* e *Viatura*, sendo que todas essas três classes devem fazer parte um mesmo *namespace*. (b) Essas duas classes, nomeadamente *Moto* e *Viatura*, deve ter cada qual um atributo protegido, respectivamente inteiro *multas* e booleano *sirene*. (c) O método *atualizar* será definido em *Moto* de maneira tal a permitir que se possa incrementar o *x* em 2.0 e o *y* em 2.0. (d) O método *atualizar* será redefinido em *Viatura* para permitir que se possa incrementar o *x* em 4.0 e o *y* em 2.0.

(Questão - 4) (a) No contexto do programa C++ em questão, crie uma classe chamada *Frota*, usando o componente *set* da STL e seu respectivo *iterator* (constante e reverso) enquanto atributos privados, ambos para endereços de objetos que possam ser apontados como *Veiculo*. [Caso se use outro componente da STL, isto permitirá fazer os demais itens nas questões, mas este presente item não será contabilizado na nota.] (b) Além de construtora apropriada, esta classe deverá ter método de inclusão apropriado, permitindo registrar apenas endereços pertinentes. (c) Além de destrutora com apropriada desalocação de memória, essa classe deverá ter método de exclusão de todos os endereços registrados (caso o conjunto não esteja vazio) chamada apropriadamente na destrutora justamente. (d) Por fim, essa classe deve ter um método de percorrimento que começa pelo último objeto apontado enquanto *Veiculo* e vai até o primeiro, chamando o método *mostrar* de cada qual.

(Questão - 5) (a) Crie uma classe *Simulador*, com um método *executar*, sendo que a instância dela (na função *main*) se chamará *simulador1*. Em *Simulador* deve ter dois objetos de *Moto* tendo os seguintes valores inicializados via construtora: para o 1º, como segue *nome* = "motociclista", *x*=2, *y*=2 e *multas* = 0; para o segundo, como segue *nome*="meliante", *x*=1, *y*=2 e *multas*= 1500. Em *Simulador* deve ter também uma instância de *Viatura*, tendo os seguintes valores inicializado via construtora *nome*="polícia", *x*=0, *y*=0 e *sirene*=true. (b) Em *Simulador* ainda haverá como atributo privado um objeto de *Frota* no qual será inserido todos os endereços de objetos pertinentes usando *cast* apropriado. (c) Todos os objetos agregados em *Simulador* devem ser alocados e desalocados dinamicamente. (d) No método *executar*, faça um laço de repetição com comando *do while* que: (1) Mostra a evolução de posição de todas os veículos via objeto pertinente de *Frota*. (2) Para quando a 'polícia' alcançar o 'meliante' ou quando o laço passar do valor de atributo de MAX iterações, sendo MAX um atributo estático constante de valor cem.