Data: 07/05/2012

Prof. Dr. Jean M. Simão - DAELN/UTFPR Curitiba/Central – PR Brasil.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus Curitiba/Central
Departamento Acadêmico de Eletrônica (DAELN). Disciplina **Fundamentos de Programação 2 - IF62C**/Turma: **S11/S12**. **Prof**: J. M. **Simão**. Curso: **Eng. Eletrônica**. **Prova sobre POO C++.**

Nome do Aluno: _____ **Turma**: ____
Horário de Começo: _____ Horário de Fim: _____

Leia toda a prova, pois os enunciados estão completados uns nos outros.
Utilize os bons princípios de Programação Orientada a Objetos (POO) em C++.

(Questão - 1) Em um programa C++ (para *console*), crie uma classe abstrata chamada de *Nave* com um único método virtual puro chamado "*void informe_natureza() = 0*" podendo, entretanto, ter outros métodos que não sejam virtuais puros, incluindo construtor e destrutor. Esta classe terá um atributo protegido inteiro chamado *número*, com seu respectivo método *get* que retorna o seu valor. Ainda, o atributo em questão não terá respectivo método *set* uma vez que ele será inicializado via parâmetro do construtor.

(Questão - 2) Crie um atributo privado (*private*), estático (*static*) e inteiro (*int*) chamado de *contador* na classe *Nave*. Este atributo *contador* será inicializado com o valor zero (0). Salienta-se que como é estático, o *contador* é inicializado fora do âmbito do método construtor na classe *Nave*. Entretanto, no construtor da classe *Nave*, o atributo *contador* será incrementado em um. Por sua vez, no destrutor da classe *Nave*, o atributo *contador* será decrementado em um. Ademais, haverá um respectivo método *get* para o atributo *contador*, sendo que tal método também será estático. Ainda, o atributo *contador* o seu respectivo método *get* não serão constantes.

Outrossim, quando pertinente, usar elementos constantes (i.e. atributos, método, retornos e parâmetros constantes) nesta classe *Nave*, assim como nas demais classes solicitadas.

(Questão - 3) Em C++, crie uma classe *Avião* (com atributo privado booleano *supersônico* e seu respectivo método *get*) e uma classe *Navio* (com um atributo privado string *origem* e seu respectivo método *get*), cada qual derivada de *Nave*, capazes de inicializar seus atributos via respectivo construtor e com um método chamado "*void informe_natureza()*". No caso de *Avião* tal método informará (em tela) "*Transporte Aéreo"*, enquanto que no caso de *Navio* informará "*Aquático"*.

(Questão - 4) Crie uma classe *Grupo_Naves* (em C++) para até 100 ponteiros de *Nave*, sendo que haverá apenas um objeto desta classe *Grupo_Naves*. Certamente, tal objeto será útil para apontar objetos de subclasses de *Nave*. Esta classe *Grupo_Naves* será implementada da forma que melhor convier ao desenvolvedor. [Por exemplo, poderia ser via um vetor (dinâmico ou não), uma lista simplesmente encadeada (dedicada ou não) ou uma lista (ou similar) baseada em componentes da *STL*, cada qual para ponteiros de *Nave*.] Ainda, a classe *Grupo_Naves* deverá ter um método para *incluir*, um método para *excluir* e outro para *listar*. A cada chamada, o método *incluir* permitirá ao objeto de *Grupo_Naves* incluir um apontamento de *Nave*. Por sua vez, o método *excluir* permitirá ao objeto de *Grupo_Naves* excluir todos os apontamentos que nele estejam. Por fim, o método *listar* permitirá ao objeto de *Grupo_Naves* apresentar a 'natureza' de cada objeto nele apontado, constituindo assim uma situação ou quadro de polimorfismo (havendo apontamentos incluídos).

(Questão - 5) Em C++, crie uma classe *Principal* que será instanciada na função *main()*. Esta classe *Principal* deve permitir agregar objetos da classe *Avião* e objetos da classe *Navio*, os quais **não** precisam ser criados dinamicamente. Outrossim, a classe *Principal* deve permitir a sua instância armazenar ponteiros de objetos *Avião* por meio de um método *void incluir_Avião (Avião* pa)*, bem como ponteiros de objetos *Navio* por meio de um método *void incluir_Navio (Navio* pn)*. Nestes métodos, entretanto, tais ponteiros serão armazenados como apontamentos de *Nave*, em um objeto de *Grupo_Naves*, utilizando-se de *static_cast*. Ainda, tais métodos serão chamados no construtor da classe *Principal*. Por fim, na *Principal* deve haver um método *executar()* onde se deve chamar o método *listar()* do objeto de *Grupo_Naves* agregado e o método *getContador()* da classe *Nave*.

A interpretação faz parte do conteúdo da prova! Inclua comentários (se for o caso) para deixar explicitas suas decisões.