



Data: 18/03/2013  
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. Dr.: Jean Marcelo 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 *Astro* com um método virtual chamado “*void informe\_natureza() { cout << “Astro sem luz” << endl; }*” podendo, entretanto, ter outros métodos (virtuais ou não), incluindo construtor e destrutor. Esta classe terá um atributo protegido string chamado *nome*, com seu respectivo método *get* que retorna o seu valor e seu respectivo método *set* que permite lhe atribuir um valor. Outrossim, quando pertinente, usar elementos constantes (i.e. atributos, método, retornos e parâmetros constantes) nesta classe *Astro*, assim como nas classes solicitadas nas demais questões.

**(Questão - 2)**

Crie um atributo privado (*private*), estático (*static*) e inteiro (*int*) chamado de *contador* na classe *Astro*. 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 *Astro*. Entretanto, no construtor da classe *Astro*, o atributo *contador* será incrementado em um. Por sua vez, no destrutor da classe *Astro*, o atributo *contador* será decrementado em um. Ademais, haverá um respectivo método *get* para o atributo *contador*, sendo que tal método também será estático.

**(Questão - 3)**

Em C++, crie uma classe *Estrela* (com atributo privado inteiro *grandeza* e seus respectivos método *get* e *set*) e uma classe *Planeta* (com um atributo privado booleano *habitável* e seus respectivos métodos *get* e *set*), cada qual derivada de *Astro*, capazes de inicializar seus atributos via respectivo construtor. No caso de *Estrela* haverá um método chamado “*void informe\_natureza()*” que informará (em tela) “*Astro com luz*”.

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

**(Questão - 4)**

(a) Crie uma classe *Grupo\_Astros* (em C++) para ponteiros de *Astro*, sendo que haverá apenas um objeto desta classe *Grupo\_Astros*. Normalmente, tal objeto será útil para apontar objetos de subclasses de *Astro*. Esta classe *Grupo\_Astros* será implementada via uma lista simplesmente encadeada (dedicada ou não) ou uma lista (ou similar) baseada em componentes da *STL*. Entretanto, em comentário no código, será necessário justificar textualmente e brevemente (em torno de cinco linhas) o porquê da abordagem escolhida.

(b) Ainda, a classe *Grupo\_Astros* 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\_Astros* incluir um apontamento de um *Astro*. Por sua vez, o método *excluir* permitirá ao objeto de *Grupo\_Astros* excluir um apontamento de uma posição especificada. Por fim, o método *listar* permitirá ao objeto de *Grupo\_Astros* apresentar o seu nome e particularmente a 'natureza' de cada objeto nele apontado, sendo necessário justificar se isto permitiria uma situação ou quadro de polimorfismo e o porquê disto.

**(Questão - 5)**

(a) Em C++, crie uma classe *Sistema\_Solar* que será instanciada na função *main()*. Esta classe *Sistema\_Solar* deve permitir agregar um objeto da classe *Estrela* (objeto Sol de grandeza 5), bem como objetos da classe *Planeta* (nomeadamente Mercúrio, Vênus, Terra, Marte, Júpiter, Saturno, Urano, Netuno e Plutão), sendo todos os objetos criados estaticamente. Outrossim, a classe *Sistema\_Solar* deve permitir a sua instância armazenar ponteiros de objetos *Estrela* por meio de um método *void incluir\_Estrela (Estrela\* es)*, bem como ponteiros de objetos *Planeta* por meio de um método *void incluir\_Planeta (Planeta\* pl)*. Nestes métodos, entretanto, tais ponteiros serão armazenados como apontamentos de *Astro*, em um objeto de *Grupo\_Astros*, utilizando-se de *static\_cast*. Ainda, tais métodos serão chamados no construtor da classe *Sistema\_Solar*.

(b) Outrossim, na classe *Sistema\_Solar* deve haver um método *executar()* onde, primeiramente, deve-se chamar o método *listar()* do objeto de *Grupo\_Astros* agregado e o método *getContador()* da classe *Astro*. Subseqüentemente no método *executar()*, deve-se chamar o método *excluir()* do objeto de *Grupo\_Astros* em questão passando a última posição, visando assim excluir o apontamento do objeto *Plutão*. Ainda no método *executar()*, deve-se chamar novamente o método *listar()* do citado objeto de *Grupo\_Astros* e o método *getContador()* da classe *Astro*. Por fim, escrever um comentário no código dizendo o valor que esta última chamada de *getContador()* retornará e o porquê deste valor.