

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba (Brasil) - Departamento Acadêmico de Eletrônica (DAELN). Disciplina: **Fundamentos de Programação 2 - IF62C/Turma: S11/S12. Prof: Jean M. Simão.**

Curso: **Engenharia Eletrônica - Prova sobre linguagem C++ – Prova de 2ª Parcial.**

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 projeto e programação orientada a objetos.**

**(Questão - 1)** Em um programa C++ (para *console*), crie uma classe abstrata chamada de *Partícula* com uma única função virtual pura podendo, entretanto, ter outras funções que não sejam virtuais puras. Ainda, esta classe terá um atributo protegido alfanumérico chamado *número*.

**(Questão - 2)** Crie uma classe *Lista\_Partículas* para ponteiros de objetos de subclasses de *Partícula* baseando-se na classe *Vector* da STL. Mais precisamente, a classe *Lista\_Partículas* deverá agregar fortemente um objeto da classe *Vector* parametrizada com (o tipo) ponteiro de *Partícula*. Ainda, a classe *Lista\_Partículas* deverá ter um método para incluir, bem como outro para listar. Por fim, a listagem apresentará a ‘natureza’ de cada objeto. Outrossim, esta classe deverá ser a única classe de lista desta prova visando uso pleno de polimorfismo.

**(Questão - 3)** Crie uma classe *Próton* (com atributo privado booleano *núcleo*), uma classe *Nêutron* (com atributo ponteiro de *Próton*) e uma classe *Elétron* (com um atributo privado inteiro *nível*), todas derivadas de *Partícula* e cada qual com uma função-membro chamada “*void informe\_natureza()*”. No caso de *Próton* tal função-membro informará (em tela) “*Próton - positivo*”, no caso de *Nêutron* informará “*Nêutron - neutro*” e no caso de *Elétron* informará “*Elétron - negativo*”.

**(Questão - 4)** Crie uma classe *Átomo*, derivada da classe *Partícula*, com uma função-membro chamada “*void informe\_natureza()*” que informará (em tela) “*Átomo*”. Cada objeto de *Átomo* poderá armazenar em si ponteiros de objetos *Elétron* por meio de uma função *void incluir\_Elétron (Elétron\* pe)*. Similarmente, cada objeto de *Átomo* poderá armazenar em si ponteiros de objetos *Próton* por meio de uma função *void incluir\_Próton (Próton \* pp)*. Entretanto, a cada apontamento de objeto *Próton* incluído, será instanciado dinamicamente um objeto de *Nêutron*, o qual será relacionado ao objeto *Próton* em questão. Ainda, cada objeto *Átomo* armazenará os apontamentos de objetos *Elétron*, *Próton* e *Nêutron* como apontamentos de objetos *Partícula* por meio de uma lista para tal

**(Questão - 5)** Crie uma classe *Molécula* onde serão criados objetos de *Átomo*, bem como objetos de *Próton* e *Elétron*. Certamente, os objetos de *Próton* e *Elétron* serão associados a objetos de *Átomo* neste âmbito. Outrossim, a classe *Molécula* terá uma lista que conterà os apontamentos destes objetos. Ainda, a classe *Molécula* terá um método *void executar()* que permitirá listar a natureza dos objetos considerados na citada lista.