

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba (Brasil) - Departamento Acadêmico de Eletrônica (DAELN). Disciplina: Fundamentos de Programação 2 - IF62C/Turma: S12. Prof: Jean M. Simão. Curso: Engenharia Industrial Elétrica, ênfase Eletrônica/Telecomunicações. Prova sobre linguagem C++ – Prova da 2ª Parcial.

Nome do Aluno: \_\_\_\_\_

Horário de Começo: \_\_\_\_\_

Horário de Fim: \_\_\_\_\_

Leia toda a prova antes de começar, pois os enunciados estão completados uns nos outros.

**(Questão 1)**

- Em um programa C++ (para *console*), além da classe *Principal*, crie também uma classe abstrata chamada de *Transporte*. Esta classe terá dois atributos: o *numero\_passageiros* e o *id*.

- Crie uma classe *Barco* e uma classe *Carro*, ambas derivadas de *Transporte*, cada qual com uma função-membro chamada “*void informe\_meio\_de\_transporte()*” que informará respectivamente “*aquático*” e “*terrestre*”. Ainda, crie pelo menos um atributo pertinente e respectivos métodos nessas classes.

**(Questão 2)** Crie uma classe *Lista\_Transp* para (ponteiros de) objetos de subclasses de *Transporte* respeitando os seguintes requisitos:

- (a) Utilizar a classe *Vector* da *STL* como essência (i.e. atributo principal) desta classe *Lista\_Transp*.
- (b) A classe *Lista\_Transp* deverá ter um método que permita incluir na ‘lista’ instanciada com *Vector*
- (c) A classe *Lista\_Transp* deverá ter um método que permita listar os elementos referenciados na ‘lista’ em questão.
- (d) A listagem apresentará o *id* de cada objeto e o seu ‘meio de transporte’.

**(Questão 3)** Faça com que a classe *Principal* tenha:

- (a) Um atributo privado que será um objeto da classe *Lista\_Transp* chamado *obj\_Lista\_Transp\_1*.
- (b) Um método para listar os objetos referenciados em *obj\_Lista\_Transp\_1*.
- (c) Um método para incluir referências de objetos em *obj\_Lista\_Transp\_1*.
- (d) Um método *menu* que permita criar objetos de *Carro* ou de *Barco* cujas referências serão mantidas em *obj\_Lista\_Transp\_1*.

Obs. gerais:

- (1) Utilizar *cout/cin* para entrada/saída, bem como *string* em vez de vetores de caractere ordinários.
- (2) Utilizar os bons princípios da orientação a objetos.

**(Questão 4)** Faça um Diagrama de Classes em *UML*, externando as classes criadas no âmbito das questões anteriores, bem como e certamente os relacionamentos entre elas.

**(Questão 5)** Responda objetivamente os seguintes questionamentos:

- (a) O que é um *Thread* em programação? Cite ainda duas *API* que dão suporte a *Threads* em C++.
- (b) O que é um *Mutex* (no âmbito de *Threads*)?
- (c) O que é um *Semáforo* (no âmbito de *Threads*)?
- (d) Explique tecnicamente (de maneira geral e suficiente) como se faz para lançar ou executar um *Thread* que executará um dado método de um objeto de uma considerada classe C++, no âmbito desta mesma classe. Em outras palavras, como usar *Thread* e C++ orientado a objetos?