Data: 08/05/2019 – 1º Semestre de 2019.
Prof. Dr. Jean M. Simão - DAINF/UTFPR Curitiba-Central.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba Central – **Dep. Acadêmico de Informática** (DAINF). Disciplina: **Técnicas de Programação – CSE20**. **Prova sobre linguagem C++ / Diagrama de Classes (*UML*) / Orientação a Objetos – Prova 1ª Parcial**

Nome do(a) Discente: _____ **Turma**: S73
Curso: *B.S.I.*_____ 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.
Em tempo, questões de mesmo peso, idem para os tópicos dentro delas (quando houver).

(Questão - 1) (A) Em um programa C++ (*console*) para simulação astronômica, crie uma classe *Astro* que terá um atributo **protegido** *string,* chamado *nome*. **(B)** Esta classe também terá um atributo **privado** *real* chamados *num_rotacao*, com seus respectivos *set* e *get*. Em tempo, nas demais classes apenas indicar a existência de *sets* e *gets*. **(C)** Ainda, essa classe deve ter um método virtual puro chamado *rotacionar* e um método normal chamado *informar_dados()* para informar valores de *nome* e *num_rotacao*. **(D)** Por fim, a classe em questão deve ter um atributo *pAstro* para ponteiro de objeto de *Astro* com respetivos métodos *set* e *get* apropriados.

(Questão - 2) (A) A classe *Astro* permitirá duas classes derivadas dela: *Estrela* e *Planeta*. Essas duas classes terão cada qual um atributo protegido, respectivamente float *grandeza* e int *posição* (em relação a uma dada estrela). **(B)** O método *rotacionar* será definido em *Estrela* de maneira tal a permitir que se possa incrementar *num_rotacao com T/grandeza*, sendo *T* uma constante real estática T, dessa classe, de valor 2,0. **(C)** O método *rotacionar* será redefinido em *Planeta* para permitir que se possa incrementar *num_rotacao* com o valor de V/*posição*, mas isto caso *pAstro* seja diferente de nulo e sabendo que V é um atributo estático inteiro não constate de valor 1. **(D)** A classe *Planeta* dever ter o operador recebimento, *void operator=*, sobrecarregado permitindo a cada objeto de *Planeta* receber um endereço de *Estrela* de forma tal que *pAstro* armazene tal valor.

(Questão - 3) (A) No contexto do programa C++ em questão, crie uma classe chamada *ConjAstros*, usando o componente *set* da STL, para endereços de objetos que possam ser apontados como *Astro*. **(B)** Esta classe deverá ter método de inclusão apropriado que não aceita mais que 30 inclusões. **(C)** Essa classe deverá ter método de exclusão de todos os elementos, sendo necessário desalocar memória como garantia para o caso dos elementos apontados eventualmente terem sido criados dinamicamente. **(D)** Por fim, essa classe deve ter um método de percorrimento que começa pelo último elemento e vai até o primeiro, o qual chama o método *rotacionar* de cada objeto apontado enquanto *Astro*. **Obs:** Até se pode usar nesta questão outro componente da STL ou mesmo tática outra, mas valendo metade da questão. Em todo caso, isso viabilizaria demais questões.

(Questão - 4) (A) Crie uma classe *SistemaSolar,* com um *método* executar, sendo que a instância dela (na função *main*) se chamará *solar*. Em *SistemaSolar* haverá como atributos três objetos (agregados) de *Planeta*, nomeadamente: (1) *mercúrio*; (2) *vênus*; e (3) *terra*. Ainda, o valor de *posição* de cada objeto de planeta está indicado junto de si nesta lista dada. **(B)** Em *SistemaSolar* haverá também como atributo uma instância de *Estrela* de quinta grandeza, chamada *sol*, que deve ser conhecida de todas as instâncias de *Planeta* via operador de recebimento**. (C)** Em *SistemaSolar* ainda haverá como atributo privado um objeto de *ConjAstros*, chamado *conj_astros*, no qual será inserido todos os endereços de objetos pertinentes usando *cast* apropriado. **(D)** No método *executar*, faça um laço de repetição com comando *for,* variando de 1 a 100.000, que mostra a evolução de *num_rotacao* de todos os astros via *conj_astros*.

(Questão - 5) (A) Primeiramente, antes das demais questões, elabore um diagrama em *UML* das classes solicitadas nelas e de seus relacionamentos com principais atributos e métodos (sendo que se deve apenas indicar construtoras, destrutoras e sets/gets, entretanto). **(B)** Nessas classes 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). **(D)** Explique sucintamente se o fato da classe *Astro* ser abstrata impediria ou não polimorfismo na solução prevista nas questões dadas.