

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba –  
Departamento Acadêmico de Informática. Disciplina: Técnicas de Programação – ICSE20.  
Exame de Substituição - C++ 2003 / Diagrama de Classes (UML) / Orientação a Objetos**

Nome do(a) Discente: \_\_\_\_\_ Turma: S71  
Curso: Eng. Comp. \_\_\_\_\_ 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++ (para *console*), crie uma classe chamada de *Ponto* com dois atributos protegidos inteiros chamados *x* e *y*. (b) Esta classe terá também um método “*virtual void imprime ( ) { draw(x, y, ' ' ) }*”. (c) Crie uma classe *Linha\_Ortogonal* derivada de *Ponto*. Esta classe terá atributos inteiros chamados *xf* e *yf*, bem como pelo menos um método, o qual deve ser “*void imprime()*”. (d) Tal método permitirá (em tela) desenhar uma linha ou semirreta ortogonal (ou seja, de “pé” ou “deitada”).

**(Questão - 2)** (a) Crie uma classe *Quadrado* derivada de *Ponto*. Esta classe terá, além do *x* e *y* derivados, um atributo inteiro *t* que determina o tamanho de cada linha do quadrado. (b) Neste sentido, a classe em questão terá ainda quatro atributos do tipo *Linha\_Ortogonal*. (c) Por fim, a classe *Quadrado* terá um método chamado “*void imprime()*”, que permitirá (em tela) desenhar um quadrado a partir do desenho das quatro linhas que a compõem. (d) Ainda, esta classe terá o operador de igual (*operator==*) sobrecarregado para que um dado *quadrado* possa verificar se está ou não coincidindo exatamente com outro *quadrado*.

**(Questão - 3)** (a) Crie uma classe *VetPontos* para ponteiros de objetos de *Ponto* por assim dizer. Na prática, a classe *VetPontos* servirá para apontar para objetos de subclasses de *Ponto*, apontados como *Ponto* bem entendido. (b) A classe *VetPontos* deve ser baseada em *STL* agregando um objeto da classe *vector* parametrizada com o tipo ponteiro de *Ponto*. (c) Ainda, a classe *VetPontos* deverá ter um método *void incluir (Ponto\* p)*, bem como outro *void percorrer()*. (d) Este método fará cada objeto apontado chamar seu método *imprime()*.

**(Questão - 4)** (a) Crie uma classe *Principal* na qual serão criados objetos do tipo *Linha\_Ortogonal*, um chamado *linha\_horizontal* e outro chamado *linha\_vertical*, sendo que tais mnemônicos retratam suas naturezas. (b) Ainda, a classe principal criará dois objetos da classe *Quadrado* cujas impressões não sejam geometricamente coincidentes, sendo que isso deve ser testado por uns dos *quadrados*. (c) Outrossim, a classe *Principal* terá um objeto de *VetPontos*, no qual os endereços destes objetos de *Linha\_Ortogonal* e de *Quadrado* serão registrados, bem como terá um método *void executar()* que permitirá desenhar tais objetos a partir de tal lista. (d) Por fim, a classe *Principal* deve respeitar o padrão de projeto singleton.

**(Questão - 5)** (a) Primeiramente, antes das demais questões, elabore um diagrama em *UML* das classes solicitadas nelas e de seus relacionamentos. (b) Defina os atributos e métodos das classes no diagrama. (c) Explique se há caso de polimorfismo previsto nas questões e onde (se for o caso). (d) Explique sucintamente para que servem padrões de projeto e cite três padrões incluindo o apresentado por si em classe, sublinhando-o.