

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba Central
Departamento Acadêmico de Informática (DAINF). Disciplina: Técnicas de Programação – CSE20
Prova sobre linguagem C++ (2003) – Exame de substituição de primeira parcialNome do(a) Discente: _____ Turma: S71
Curso: _____ Horário de Começo: _____ Horário de Fim: _____

Leia toda a prova, pois os enunciados estão completados uns nos outros. Ainda, utilize os bons princípios de programação orientada a objetos. Por fim, questões de mesmo peso, idem para os tópicos dentro delas.

(Questão - 1) (a) Em um programa C++ (para *console*), crie uma classe chamada de *Ponto* com atributo *OK* inicializado com *true*. (b) Esta classe deve ter ainda dois atributos inteiros chamados *x* e *y*. (c) Essa classe terá também um método “*virtual void imprimir () const { draw(x, y, '.'); }*” sendo que o comando ali usado desenha um ponto nas coordenadas dadas por parâmetro. (d) Ainda, esta classe terá o operador de igual sobrecarregado para que um dado *ponto* possa verificar se está ou não coincidindo com outro *ponto*.

(Questão - 2) (a) Crie uma classe *VetPontos* que servirá tanto para apontar para objetos das subclasses de *Ponto* quanto objetos da classe *Ponto* em si, todos apontados como *Ponto* bem entendido. (b) A classe *VetPontos* deve ser baseada em *STL* tendo um objeto da classe *vector* parametrizada com o tipo ponteiro de *Ponto*, bem como ter um método *const int getTamanho() const*. (c) Ainda, a classe *VetPontos* deverá ter um método *void incluir (Ponto* p)* para inclusão de endereços pertinentes. (d) Por fim, ela terá também um método *void percorrer() const* que faz cada objeto apontado chamar seu método *imprimir()*.

(Questão - 3) (a) Derivada de *Ponto*, crie uma classe *LinhaOrtogonal* que tenha atributos inteiros chamados *xf* e *yf* (sendo que o *f* denota final). (b) Ainda, *LinhaOrtogonal* agrega um conjunto de pontos via *VetPontos* caso haja ortogonalidade a partir das coordenadas cartesianas dadas, deixando *OK* com valor *true*. Caso contrário, *OK* deve ter valor *false*. (c) A classe *LinhaOrtogonal* terá também um método “*void imprimir() const*” que permitirá (em tela) desenhar uma linha ou semirreta ortogonal via objeto de *VetPontos*. (d) Mais precisamente, tal método permitirá desenhar uma semirreta ortogonal (ou seja, de “pé” ou “deitada”) ou, senão (em caso de não ortogonalidade) nada desenhará.

(Questão - 4) (a) Derivada de *Ponto*, crie uma classe *Retangulo*. (b) Esta classe terá, além do *x* e *y* derivados, um atributo inteiro *A* que determina a altura de cada linha do retângulo, bem como outro atributo inteiro *L* que determina a largura. (c) Ademais e certamente, a classe em questão terá ainda quatro atributos do tipo *LinhaOrtogonal* sendo que eles determinarão se seu atributo *OK* será *true* ou *false*. (d) Por fim, a classe *Retangulo* terá um método “*void imprimir() const*”, que permitirá desenhar um retângulo a partir do desenho das suas quatro linhas, caso *OK* seja *true*.

(Questão - 5) (a) Crie uma classe *Principal* que deve agregar dois objetos de *Ponto*, bem como um objeto da classe *LinhaOrtogonal* chamado de *linha_ao_ortogonal* sendo que tal mnemônico retrata sua natureza. A classe *Principal* também deve agregar um objeto da classe *Retangulo*. Todos estes agregados devem ser criados dinamicamente e, cada qual, deletado ainda na construtora caso *OK* for *false*. (b) Ainda, a classe *Principal* terá um objeto *conjunto* (criado estaticamente) da classe *VetPontos*, no qual os endereços destes objetos agregados serão registrados/apontados caso *OK* esteja com valor *true*. (c) Ademais, a classe em questão terá um método *void executar()* que permitirá desenhar tais objetos a partir de *conjunto*, bem como exercitar (diretamente ou indiretamente) todas as funções membros (salvo construtoras, destrutoras e *gets* e *sets*) de seus agregados diretos e criados dinamicamente, via *conjunto* e/ou de outra forma. (d) Por fim, a classe *Principal* deverá respeitar o padrão de projeto *singleton*.