

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)**  
Campus de Curitiba-PR (Brasil)  
DAELN - Departamento Acadêmico de Eletrônica.

Curso: **Engenharia Industrial Elétrica, ênfase Eletrônica/Telecomunicações.**

Disciplina:

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)** Considere uma classe não abstrata chamada *Geometrico* que terá dois atributos inteiros nomeados como  $x$  e  $y$ . Estes atributos devem ser (diretamente) acessíveis somente no escopo de sua classe e de suas derivadas. Na verdade, *Geometrico* servirá de base elementar para demais classes, tais como *Ponto*, *RetaOrtogonal*, *Retângulo* e *Quadrado*. Estas classes se relacionarão entre si permitindo a elaboração de um editor de figuras elementar. Assim sendo, construa um diagrama de classes da UML que expresse estas classes citadas e outras duas classes de sua escolha (certamente para fins geométricos), bem como os relacionamentos modelados nestas classes.

**(Questão - 2)** A classe *Ponto* deve ter um método chamado *Desenha* cuja implementação permitirá imprimir um ponto em uma coordenada  $x$  e  $y$  da tela. A classe *RetaOrtogonal*, por sua vez, deve ter um método também chamado *Desenha* cuja implementação permitirá imprimir um segmento de reta ortogonal ao eixo  $x$  ou ao eixo  $y$ .

Estas duas classes ainda deverão comportar o seguinte relacionamento: um objeto *RetaOrtogonal* será formado por uma lista de objetos *Ponto*, cada um tratando um ponto que forma ('tecnicamente') o segmento de reta naquele objeto. Isto é determinado supondo-se utilidades futuras para todos estes objetos *Ponto* e não somente para aqueles que representam as extremidades do segmento de reta.

Uma vez que o modelo de classes esteja pronto, desenvolva então o código C++ orientado a objetos equivalente, atendendo estas determinações (acima) e as boas práticas de orientação a objetos. Neste sentido, visando reaproveitamento, implemente o relacionamento citado entre *Ponto* e *RetaOrtogonal* (e mesmo outros que se justifiquem) usando o *template* Lista desenvolvido em sala.

Obs.1: Para implementar o método *Desenha* da classe *Ponto*, será permitido o uso do *gotoxy(x,y)* da *conio.h* que posiciona o cursor em um ponto da tela, considerada como 80 por 60 pontos. Normalmente, usa-se *gotoxy* seguido de *printf*.

Obs.2: Pelos menos os métodos *Desenha* das classes *Ponto* e *RetaOrtogonal* devem estar implementados cf. especificado. Este último permitirá avaliar a capacidade de implementar os relacionamentos definidos utilizando o *template* Lista.

Exemplo de Prova  
Prof. J. M. SIMÃO

**(Questão - 3)** Segundo o purismo de Orientação a Objetos, doutrinado nesta disciplina, o código C++ deverá ter uma classe *Principal* que pode (e normalmente deve) estar modelada no respectivo diagrama UML. Neste código em questão, a classe *Principal* disponibilizará um método que fornecerá as opções de desenho para o usuário (i.e. um *menu*) e que chama um outro método que desenha uma figura na tela conforme a opção do usuário, desde que a opção seja válida e não seja ‘sair do aplicativo’.

Sabendo-se que é um aplicativo “console”, utilize apenas a biblioteca *iostream* neste método *Menu*. Este método ainda será responsável por: (a) informar ao usuário que ele poderá re-invocar o *menu* utilizando a tecla ‘M’, (b) chamar um método que apaga, antes de imprimir o menu, as figuras que tenham sido desenhadas anteriormente; (c) limpa o menu assim que uma opção tenha sido escolhida; e (d) chama um método que *redesenha* as figuras preexistentes antes de desenhar a nova figura escolhida.

Os métodos que desenharam e apagam figuras estarão na classe *Principal*. Cada um deles acessará as figuras via uma lista e as desenhará ou apagará cf. seu objetivo. Entretanto, a classe *Principal* será implementada apenas como uma lista de objetos de *Geometrico*, assim sendo cada objeto criado terá que ser apropriadamente armazenado nela e seu método de *redesenhar* ou de *apagar* (cf. o caso) invocado a partir do objeto recuperado desta lista.

**(Questão - 4)**

Considerando o ‘aplicativo desenvolvido’ acima, identifique o conjunto de eventos que ocorrem nele e qual é o conjunto de métodos que tratam estes eventos.