

Curso: Engenharia Industrial Elétrica, ênfase Eletrônica/Telecomunicações.  
Disciplina: Fundamentos de Programação 2 – IF62C – Turma: S11.

Nome do Aluno: \_\_\_\_\_  
Horário de Começo: \_\_\_\_\_ Horário de Fim: \_\_\_\_\_

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

[ valor 20% ] (1) Atendendo as boas práticas de orientação a objetos, resolva os seguintes tópicos em C++:

(a) Elabore uma classe abstrata chamada *Geometrico* (com dois atributos inteiros *x* e *y* protegidos) que servirá de base (direta ou indiretamente) para as classes *Ponto*, *RetaOrtogonal* e *TrianguloRetangulo*.

(b) Elabore uma classe *Ponto* com:

- um método chamado *desenha* que permitirá imprimir um ponto em uma coordenada da tela<sup>1</sup>; e com
- o operador de igualdade sobrecarregado permitindo comparar os valores dos atributos de dois pontos.

[ valor 30% ] (2) Atendendo as boas práticas de orientação a objetos, resolva os seguintes tópicos em C++:

(a) Elabore uma classe *RetaOrtogonal* com um método chamado *desenha* que permitirá imprimir um segmento de reta ortogonal ao eixo x ou ao eixo y. Cada objeto *RetaOrtogonal* será tecnicamente formado por objetos *Ponto*, visando potencial uso deles. Implemente este relacionamento usando o *vector* da STL.

(b) Elabore uma classe *TrianguloRetangulo* que será composto por dois catetos (que serão objetos de *RetaOrtogonal*) e uma hipotenusa (que pode ser uma instância de *vector* parametrizado com *Ponto*). Esses catetos serão de tamanho igual, um paralelo ao eixo x e outro paralelo ao eixo y. Certamente, esta classe terá um método chamado *desenha* para desenhar o triângulo retângulo.

[ valor 20% ] (3) Atendendo as boas práticas de orientação a objetos, resolva os seguintes tópicos em C++:

(a) Elabore uma classe *Principal* com:

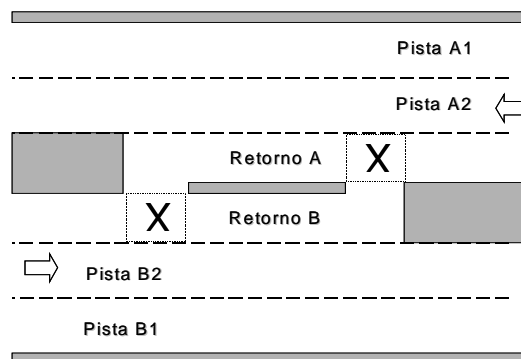
- um método que fornecerá as opções de desenho para o usuário (i.e. um *menu*) e
- uma lista para endereços de objetos de Geométricos criada com o *list* da STL.

(b) Considere ainda que:

- cada vez que o menu é impresso, as figuras são apagadas; e que (depois da escolha da figura)
- tanto a figura escolhida quanto as que estavam previamente impressas deverão ser apresentadas em tela.

[ valor 20% ] (4) Modele as classes C++ elaboradas nas questões anteriores, bem como suas relações, por meio de um diagrama de classe da UML (*Unified Modeling Language*).

[ valor 10% ] (5) Considere a figura abaixo (que vem a ser a essência de um exercício explicado em sala). Em suma, há carros que navegam nas pistas A (em um sentido) e carros que trafegam nas pistas B (em outro sentido). De tempos em tempos, alguns desses carros tomam o retorno resultando em um tráfego no sentido contrário do qual vinham. Certamente, nos retornos, as posições X não devem ser bloqueadas. Assim sendo, de maneira objetiva, qual seria uma solução viável para simular este cenário usando *threads*?



<sup>1</sup> Para tal, é permitido o uso do comando *Console::SetCursorPosition ( x , y )* que posiciona o cursor em um ponto da tela (80 por 60 pontos). Usa-se este comando seguido de *cout*.