



Universidade Federal de Uberlândia
Curso de Graduação em Gestão da Informação
1º Trabalho de Estruturas de Dados I – Prof. Daniel A. Furtado
Trabalho Individual – Introdução à Programação Orientada a Objetos

INTRUÇÕES GERAIS

- Esta atividade deve ser realizada individualmente;
- Este trabalho deve ser feito utilizando a linguagem C# ou Python. Trabalhos implementados utilizando outras linguagens serão anulados;
- **Todo o código deve ser colocado em um único arquivo**, pois apenas um arquivo (.cs ou .py) deve ser entregue. Não crie arquivos adicionais para as classes em C#;
- Não compactar os arquivos no momento da entrega;
- Os recursos adequados das linguagens devem ser utilizados;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet ou de trabalhos de semestres anteriores serão anulados;
- As implementações não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato, etc.;
- O trabalho deve ser entregue até a data definida pelo professor em aula síncrona;
- Trabalhos enviados por e-mail **não serão considerados** (veja instruções no final);

Exercício A) Criar uma classe para modelar um ponto geométrico no espaço, com coordenadas x, y e z. A classe deve possuir atributos e métodos conforme especificação a seguir:

- As coordenadas x, y e z devem ser atributos privados, do tipo *float*;
- A classe deve ter um construtor contendo três parâmetros correspondentes às coordenadas x, y e z, que devem ser utilizados para inicializar os respectivos atributos;
- A classe deve possuir métodos públicos do tipo 'getters' que permita acesso às coordenadas x, y e z;
- A classe deve possuir um método para calcular a distância entre dois pontos, onde um deles corresponde ao próprio objeto e o segundo é fornecido como parâmetro. Tal método deve possibilitar o uso conforme apresentado a seguir:

```
MeuPonto3D p1 = new ...  
MeuPonto3D p2 = new ...  
  
double dist = p1.DistanciaAte(p2);
```

Para calcular a distância, utilize a fórmula a seguir:

$$Dist_{P_1P_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Exercício B) Criar uma classe para modelar um vetor geométrico no espaço R^3 , com coordenadas A , B e C . A classe deve possuir atributos e métodos conforme especificação a seguir:

- A classe ponto solicitada no item anterior **não deve** ser utilizada nesta modelagem;
- As coordenadas A , B , C devem ser atributos públicos, do tipo *float*;
- A classe deve ter um construtor contendo três parâmetros correspondentes às coordenadas do vetor, para criação e inicialização dos objetos;
- A classe deve possuir um método para calcular e retornar o módulo (ou norma) do vetor. O módulo de um vetor v em R^3 pode ser calculado pela fórmula a seguir, onde A , B e C são as coordenadas espaciais do vetor:

$$|v| = \sqrt{A^2 + B^2 + C^2}$$

- A classe deve possuir um método que permita normalizar o vetor (calcular um vetor unitário na mesma direção). Para normalizar o vetor, basta recalculá-lo dividindo cada uma das coordenadas pelo módulo do vetor (atenção: o método não deve retornar um novo vetor normalizado, mas sim alterar os atributos do próprio objeto). Informações básicas sobre vetores podem ser encontradas em:

<http://www.uel.br/projetos/matessencial/basico/geometria/vetor3d.html>

Exercício C) Criar uma classe para modelar um plano geométrico no espaço R^3 . A classe deve possuir atributos e métodos conforme especificação a seguir:

- Um plano pode ser representado por meio de um ponto (pertencente ao plano) e um vetor normal ao plano. Portanto, a classe deve possuir dois atributos correspondentes ao ponto e ao vetor normal (utilize as definições anteriores de ponto e vetor);
- A classe deve possuir um construtor contendo dois parâmetros correspondentes ao ponto e ao vetor normal, para criação e inicialização dos planos;

Exercício D) Crie um “código principal” que exemplifique a utilização de todas as classes e respectivos métodos definidos anteriormente.

Entrega

O arquivo principal contendo as definições das classes e o programa principal deve ser enviado pelo Sistema de Aplicação de Testes (SAAT), até a data limite indicada pelo professor. Não envie arquivos compactados, ou outros arquivos do Visual Studio que não sejam o arquivo de código principal.

Sobre Eventuais Plágios

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.