



Universidade Federal de Uberlândia

Faculdade de Computação – Gestão da Informação – Prof. Daniel A. Furtado

5º Trabalho de Programação Orientada a Objetos

Propriedades e Membros Estáticos

Instruções Gerais

- Esta atividade deve ser realizada **individualmente**;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet, gerados por ferramentas de IA (como ChatGPT, Gemini e similares), copiados de trabalhos de semestres anteriores, serão anulados;
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados a eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

Material de Apoio

<https://furtado.prof.ufu.br/site/teaching/POO/POO-Modulo2-Fundamentos.pdf> (slides 56-81)

Exercício 1

Faça uma cópia do código C# desenvolvido para o **Exercício 1** do **Trabalho 4**. Substitua os métodos **GetVelocidade** e **SetVelocidade** por uma propriedade de nome **Velocidade**. A propriedade deve continuar garantindo as funcionalidades dos métodos anteriores. Atualize o programa principal para a propriedade seja acessada no lugar dos métodos.

Exercício 2

Crie um projeto no Visual Studio Community e defina uma classe para modelar um **Retângulo** conforme especificações a seguir:

- Deve haver atributos **privados** para armazenar a base e a altura do retângulo (**double**);
- O construtor deve permitir a criação de retângulos informando a respectiva base e altura;
- Crie as propriedades públicas **Base** e **Altura** para fornecer acesso indireto aos atributos base e altura do retângulo. As propriedades não devem permitir valores nulos ou negativos;
- Crie uma propriedade computada que retorne a **área** do retângulo. Utilize o operador **=>**;
- Crie um **atributo estático count** para contabilizar o número de objetos criados da classe. Faça as devidas alterações no construtor;
- Crie uma **propriedade estática Count** para expor o atributo **count** para leitura;
- Crie um método estático que receba **dois** objetos do tipo **Retângulo** como parâmetro e retorne **verdadeiro** ou **falso** indicando se os retângulos são iguais ou não (comparando as bases e alturas).

Crie um "programa principal" ilustrando o uso da classe e das propriedades. O programa deve criar dois retângulos com valores de base e altura informados pelo usuário. No final, deve mostrar ao usuário o valor da base, da altura e da área de cada retângulo criado. Deve mostrar também o número de instâncias criadas utilizando a propriedade estática **Count** e se os dois retângulos são iguais ou não (utilizando o método estático de comparação de retângulos).

Exercício 3

Faça uma cópia do código C# desenvolvido para o **Exercício 2** do **Trabalho 4**. Substitua todos os métodos do tipo getters/setters por respectivas **propriedades**. Utilize o operador `=>` sempre que possível.

Entrega

Compacte as pastas dos projetos dos exercícios em um único arquivo no formato **zip** e envie pelo Sistema **SAAT** até a data limite indicada pelo professor em sala de aula. Caso não tenha o **Visual Studio Community**, coloque o código criado para cada exercício em um arquivo .cs (exercicio1.cs, exercicio2.cs etc.), compacte todos os arquivos e envie o arquivo compactado.

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**.