



Universidade Federal de Uberlândia  
Faculdade de Computação – Gestão da Informação – Prof. Daniel A. Furtado  
6º Trabalho de Programação Orientada a Objetos  
Agregação, Referência para Objetos, Garbage Collector

### 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 69-96)

### Exercício 1

---

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

- Deve haver atributos **privados** para armazenar o **cep**, a **rua** e a **cidade**;
- O construtor deve permitir a criação de novos endereços informando o cep, a rua e a cidade;
- Crie um **método** para exibição dos dados do objeto;
- Crie uma propriedade para expor o atributo cidade para leitura e escrita.

Defina outra classe para modelar **clientes** conforme descrição a seguir:

- Deve haver atributos privados do tipo **string** para armazenar o **CPF** e o **nome** do cliente;
- Deve haver atributos privados para armazenar o endereço residencial do cliente e o endereço de trabalho. Utilize a classe **Endereco** definida anteriormente;
- Deve haver um construtor que permita a criação de novos objetos informando o CPF, o nome e os endereços (residencial e trabalho);
- Crie um método para exibição dos dados do cliente.

Crie um "programa principal" ilustrando o uso das classes. Crie dois objetos da classe **Cliente**: um com endereços de casa e trabalho distintos e outro com endereços iguais (utilizando o mesmo objeto do tipo endereço). Em seguida, altere a cidade do endereço **residencial** para os dois clientes. Apresente todos os dados dos clientes e observe as mudanças.

### Exercício 2

---

- a) Modifique a classe **Endereco** do exercício anterior. Adicione um método de nome **Clonar** que retorne um novo objeto do tipo **Endereco** contendo exatamente os mesmos dados do objeto corrente. Mostre, no programa principal, que o método realmente produziu um clone do objeto original (que não é apenas uma cópia da referência).

- b) Modifique a classe **Cliente** do exercício anterior. Adicione um método que faça a clonagem total do objeto. O método deve retornar um novo objeto contendo os dados clonados (incluindo os endereços). Mostre, no programa principal, que o método realmente produziu um clone do objeto original (que não é apenas uma cópia da referência).
- c) Modifique o **método** de clonagem da classe **Cliente** para que seja contabilizado o número total de clientes clonados por meio do método. Exemplifique o uso no programa principal.

### Exercício 3

---

- a) Explique o funcionamento geral do Garbage Collector (GC) em linguagens como Java e C#.
- b) Explique o propósito do método **destrutor** em C#. Explique como e quando ele é chamado.
- c) Considere o código a seguir. O que acontece com o ponto (0,0,0) em memória depois que a segunda linha do código é executada?

```
var p1 = new Ponto3D(0, 0, 0);  
p1 = new Ponto3D(1, 1, 1);
```

### Entrega

---

Compacte os arquivos com as respostas 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**.