



Universidade Federal de Uberlândia
Faculdade de Computação – Gestão da Informação – Prof. Daniel A. Furtado
11º Trabalho de Programação Orientada a Objetos
Polimorfismo, Ligação Dinâmica, Interfaces

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>

<https://furtado.prof.ufu.br/site/teaching/POO/POO-Modulo3-Heranca.pdf>

Exercício 1

- a) Defina uma classe **abstrata** para modelar **animais** de uma forma geral. A classe deve ter uma propriedade pública concreta para armazenar o **nome** do animal e um método abstrato **EmitirSom()**.
- b) Defina duas classes **especializadas** (que herdem da classe Animal) para modelar, respectivamente, **cães** e **gatos**. Crie uma propriedade específica de sua escolha em cada classe. O construtor deve ter um parâmetro para inicialização do **nome** e da propriedade específica. Cada classe deve ter sua própria definição do método **EmitirSom()** que apresente na tela uma mensagem correspondente ao som do respectivo animal.
- c) Crie um “programa principal” ilustrando o uso das classes. O programa deve criar um array de objetos do tipo **Animal** e percorrer o array utilizando um laço **foreach** para apresentação do **nome** e emissão do respectivo som de cada objeto do array.
- d) Explique os conceitos de **polimorfismo** e **ligação dinâmica** utilizando como base o exemplo implementado.

Exercício 2

- a) Defina uma interface **ILocalizavel** que contenha a definição do método **CoordenadaGPS**, sem parâmetro, com retorno em **string**.
- b) Defina uma classe abstrata **Pessoa** que contenha uma propriedade **Nome** e um construtor para sua inicialização. A classe deve possuir também uma definição de método abstrato **CalcularRendimentoAnual()** com retorno em **double**.
- c) Defina uma classe concreta **PessoaFisica**, que herde de **Pessoa** e implemente a interface **ILocalizavel**. A classe deve possuir uma propriedade específica **CPF** e um construtor para

correta inicialização. Para calcular o rendimento anual, considere que toda pessoa física possui rendimento mensal de 5 salários mínimos. O método de cálculo da localização deve retornar a coordenada fictícia "lat -18.9, long -48.2".

- d) Defina uma classe concreta **Automovel** que contenha as propriedades **placa** (string), **modelo** (string) e **ano** (int) e um construtor para inicialização das propriedades. A classe deve implementar a interface **ILocalizavel**. Considere que todos os veículos estejam localizados na coordenada "lat -15.7, long -47.8".
- e) Crie um "programa principal" para ilustrar o uso das classes, métodos e interface. O programa deve ilustrar o conceito de polimorfismo e ligação dinâmica.
- f) Explique os seguintes conceitos:
 - i. Classe abstrata x classe concreta
 - ii. Classe concreta x classe estática
 - iii. Método abstrato de uma classe abstrata
 - iv. Método de instância x método de classe (estático)
 - v. Classe abstrata x Interface

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, juntamente com as explicações, 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**.