Universidade Federal de Uberlândia
Faculdade de Computação – Gestão da Informação – Prof. Daniel A. Furtado
9º Trabalho de Programação Orientada a Objetos
Introdução à Herança

Instruções Gerais

- Esta atividade deve ser realizada **individualmente**;
- Esteja atento às **observações sobre plágio** apresentadas no final desde 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 à 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-Modulo3-Heranca.pdf (slides 1-14)

Exercício 1

a) Defina uma classe para representar pessoas de uma forma geral, seja pessoa física ou pessoa jurídica. A classe deve ter uma propriedade pública **Nome** e um construtor que permita a criação de novos objetos da seguinte forma: `var pessoa1 = new Pessoa("Fulano de Tal");`

b) Defina classes derivadas para representar pessoas físicas e jurídicas. Devem conter, respectivamente, um **CPF** e um **CNPj**. As classes devem ter construtores que permitam a criação de objetos da seguinte forma:
   - `var pessoaFisica1 = new PessoaFisica("Fulano de Tal", "cpf123");`
   - `var pessoaJuridica1 = new PessoaJuridica("Empresa X", "cnpj123");`

c) Crie um programa principal exemplificando o uso de todas as classes e de suas propriedades.

Exercício 2

a) Defina uma classe base chamada **Conta**. Ela deve conter as seguintes propriedades públicas:
   - **Titular** (string)
   - **Numero** (int)
   - **Agencia** (string)

   O construtor deve permitir a criação de novos objetos passando esses três parâmetros. Exemplo: `var conta1 = new Conta("João", 123, "0001-X");`

b) Defina as classes derivadas **ContaPoupanca** e **ContaCorrente**. A ContaPoupanca deve ter a propriedade adicional **Variacao** (int). A ContaCorrente deve ter a propriedade adicional **Limite** (double). Os construtores das classes filhas devem receber todos os dados (incluindo os da classe base) e repassá-los corretamente. Exemplo:
   - `var contaPoupanca1 = new ContaPoupanca("Fulano de Tal", 456, "0002-X", 1);`
   - `var contaCorrente1 = new ContaCorrente("Empresa A", 789, "0003-X", 1000.00);`

c) Crie um programa principal que instancie um objeto de cada classe, sem solicitar os dados ao usuário, e exiba no console os dados de cada conta, mostrando que as classes derivadas herdaram e preencheram corretamente as propriedades **Titular**, **Numero** e **Agencia**;
d) Acrescente o devido código no programa principal para que duas contas adicionais sejam criadas (uma conta poupança e uma conta corrente) com dados fornecidos pelo usuário. Acrescente um método **Mostrar** na classe mais geral (Conta) para exibição do titular da conta, do número e da agência na janela de console. Em seguida, utilize esse método para que os dados das duas contas adicionais sejam apresentados ao usuário.

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**.