



Universidade Federal de Uberlândia
Faculdade de Computação – Gestão da Informação – Prof. Daniel A. Furtado
10º Trabalho de Programação Orientada a Objetos
Classes Abstratas, Métodos Virtuais e Abstratos, Sobrescrita de Métodos

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-Modulo3-Heranca.pdf> (slides 1-28)

Exercício 1

- a) Crie uma classe base de nome **Funcionario**. A classe deve possuir:
 - Duas propriedades públicas: Nome (string) e Salario (double).
 - Um construtor para inicializar as propriedades.
 - Um método **virtual** chamado **CalcularBonificacao()** que retorne o valor de 10% do salário.
- b) Crie uma classe derivada de nome **Gerente**. A classe deve sobrescrever o método **CalcularBonificacao()** da classe base para que seja retornado uma bonificação maior, correspondente a 20% do salário. Deve possuir também uma propriedade específica **LimiteCompra** (double).
- c) Crie uma classe derivada de nome **Diretor**. A classe deve sobrescrever o método **CalcularBonificacao()** da classe base para que seja retornado uma bonificação ainda maior, correspondente à bonificação base de um funcionário comum (chamar o método **CalcularBonificacao()** da classe base) acrescido de mais 20% do salário.
- d) Crie um programa principal exemplificando o uso de todas as classes, métodos e propriedades.

Exercício 2

- a) Crie uma classe **abstrata** de nome **Pagamento** contendo uma propriedade para armazenar o **valor original** do pagamento. Deve haver um construtor para inicialização da propriedade e um método **abstrato** **CalcularTotalAPagar()**, com retorno em **double**, que obrigue as classes derivadas a definirem eventuais descontos ou acréscimos para cálculo do valor final a ser pago.
- b) Crie uma classe derivada concreta de nome **PagamentoPix**. A classe deve implementar o método **CalcularTotalAPagar()** aplicando um desconto de 10% sobre o valor original do pagamento (multiplique o valor original por 0.9). O valor descontado deve ser retornado. Deve haver um construtor adequado.

- c) Crie uma classe derivada concreta de nome **PagamentoCredito** que contenha uma propriedade própria de nome **NumeroDeParcelas** (int) e um construtor para inicialização das propriedades. A classe deve implementar o método **CalcularTotalAPagar()** acrescentando juros simples de 2% ao mês ao valor original, conforme número de parcelas ($\text{valor total} = \text{valor original} * (1 + (0.02 * \text{número de parcelas}))$).
- d) Crie um programa principal ilustrando o uso das classes, métodos e propriedades.
- e) Adicione o código adequado ao programa principal (sem remover o código do item anterior) para que seja solicitado ao usuário o **valor original** do pagamento. Em seguida, o programa deve solicitar ao usuário a forma de pagamento (1 – Pix, 2 – Crédito). Se o usuário escolher a opção 2, deve ser solicitado também o número de parcelas. Na sequência, o programa deve instanciar um objeto da classe apropriada e mostrar ao usuário o valor final a ser pago.

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**.