Linguagem Orientada a Objetos

Encontro 04 – Diagrama de Classe, Classes e Objetos – Parte 2

Prof. Luiz Augusto Rodrigues
luiz.a.rodrigues@cogna.com.br

Avisos Importantes

Nenhum material será enviado via e-mail. Os materiais serão disponibilizados no **AVA** e no google drive: **https://bit.ly/3rJL6o3**

Dúvidas, questionamentos, entre outros deverão ser realizados pelo **e-mail** e pelo **Whatsapp** da disciplina**.**

Para ingressar no grupo do **Whatsapp** da disciplina acesse o link a seguir e selecione sua disciplina. **linklist.bio/profluizao_2023-2**

A Disciplina de Programação Orientada a Objetos, a partir deste slide, será referenciada pela sigla **POO**.

Assuntos Abordados

- Revisão – Pilares
- Correção da Atividade 03
- Classes Abstratas
- Classes Estáticas
- Conteúdo Prático
  - Otimizando código.

LOO

Encontro 04

Instruções Importantíssimas!!!

Instruções Importantíssimas!!!

Para este semestre, na disciplina de POO, utilizaremos OBRIGATORIAMENTE as seguintes ferramentas:

- Sistema Operacional Windows 10, Linux ou MacOS.
- Linguagem de Programação Java.
- JDK 17 ou superior
  - https://download.oracle.com/java/17/archive/jdk-17.0.6_windows-x64_bin.exe
- VS Code.

Instruções Importantíssimas!!!

- Git 2.39.2 ou superior
  - https://github.com/git-for-windows/git/releases/download/v2.39.2.windows.1/Git-2.39.2-64-bit.exe
- Github
  - https://github.com/
  - https://desktop.github.com/
- Tutorial em Git e Github
  - Tutorial Git e Github 2022 – Introdução prática para iniciantes
  - Canal DevSuperior
  - https://youtu.be/_hZf1teRFNg

Quem for utilizar Linux ou MacOSx, fale comigo após a aula, ou pelo grupo do WhatsApp, para instruções de instalação.

Instruções Importantíssimas!!!

- Os relatórios não serão utilizados nessa disciplina, por se tratar de uma disciplina 90% prática.
- No lugar, teremos atividades de pós-aula, que deverão ser respondidas e justificadas, para fixação do conteúdo discutido na aula.
- As atividades de aula estarão disponíveis apenas no Github da turma.
- O tutorial do Git e Github é **obrigatório**.

LOO

Encontro 04

Revisão – Pilares

Princípios de POO

LOO

Encontro 04

Correção da Atividade 03

Correção da Atividade 03

URL para a Atividade 03:

https://drive.google.com/file/d/1TySS_WtWDXqunhrqxelNxBEgMOmSoG_Z/view?usp=drive_link

LOO

Encontro 04

Classes Abstratas

Classes Abstratas

De acordo com a **Microsoft Docs**:

O modificador abstract indica que o item que está sendo modificado tem uma implementação ausente ou incompleta. Pode ser usado com classes, métodos, propriedades, indexadores e eventos.

Use o modificador abstract em uma declaração de classe para indicar que uma classe se destina somente a ser uma classe base de outras classes, não instanciada por conta própria.

Membros marcados como abstratos precisam ser implementados por classes não abstratas que derivam da classe abstrata.

Classes Abstratas

Uma classe abstrata é uma classe que serve de modelo para outras classes. Ela sempre será uma superclasse genérica, e suas subclasses serão mais específicas.

Além disso, ela não pode ser instanciada e pode conter ou não métodos abstratos, podendo ser implementados nas classes descendentes.

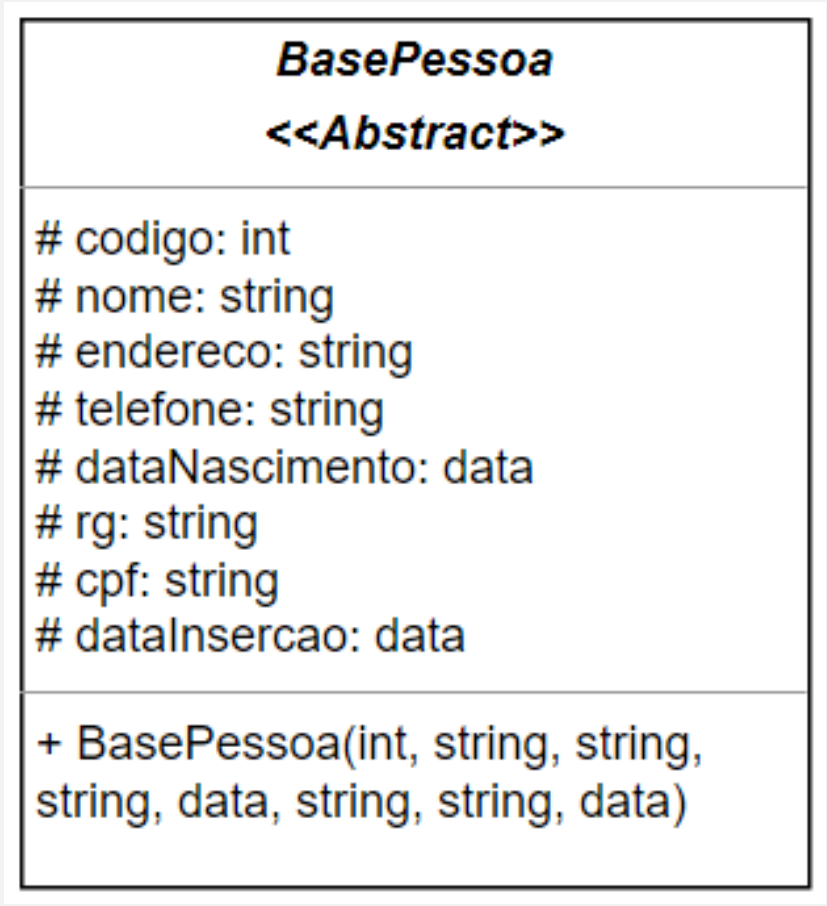
Ou seja, uma classe abstrata pode implementar ou não um método, sendo obrigatória a existência de pelo menos um método abstrato, sem corpo.

Ao ouvirmos a palavra animal, podemos imaginar um gato, um cachorro, etc. Ou seja, uma palavra genérica pode nos levar à imaginar coisas mais específicas.

Classes Abstratas - Características

- Uma classe abstrata é uma classe que não pode ser instanciada. Você não pode criar um objeto a partir de uma classe abstrata.
- Uma classe abstrata pode ser herdada e geralmente serve como classe base para outras classes.
- Uma classe abstrata pode conter métodos abstratos e métodos comuns. Também podem possuir construtores, propriedades, indexadores e eventos.
- Uma classe abstrata não pode ser estática (static).
- Uma classe abstrata não pode ser selada (sealed)(*final*).
- Uma classe abstrata pode herdar de outra classe abstrata.

Classes Abstratas - UML



- As Boas Práticas recomendam o uso do prefixo ***Base*** , em situações comuns. Deve-se levar em conta a camada na qual a classe existe.

Métodos Abstratos

Um método indicado pela palavra abstract indica que o mesmo não possui código implementado, e deve ter sua implementação definida na próxima classe real da hierarquia.

- Um método abstrato é um método que não possui implementação na classe abstrata.
- Possui somente a definição de sua assinatura. A sua implementação deve ser feita na classe derivada.
- É por natureza um método virtual e deve ser implementado usando o modificador override.
- Somente pode existir em uma classe abstrata.
- Não pode usar os modificadores ***static*** e ***virtual***.

Propriedades Abstratas

Uma declaração de propriedade do tipo ***abstract*** não fornece uma implementação propriedade – ela declara que a classe dá suporte às propriedades, mas deixa a implementação para classes derivadas.

Ou seja, nós simplesmente declaramos sua assinatura. O código de implementação deve obrigatoriamente ser desenvolvido na classe real mais próxima na hierarquia de herança.

LOO

Encontro 04

Classes Estáticas

Classes Estáticas

Objetos estáticos são objetos que possuem apenas uma instância e são definidos quando utilizamos a palavra-chave ***static***.

Quando declaramos um objeto como estático, significa que não importa quantos deles sejam criados, existirá apenas uma cópia dele na memória. Ou seja, existe apenas uma instância do objeto.

Uma classe estática é basicamente igual a uma classe não-estática. ***Porém uma classe estática não pode ser instanciada***. Isto quer dizer que não podemos utilizar a palavra-chave ***new*** para criá-la.

Por este motivo, você pode acessar os membros de uma classe estática apenas usando o nome da classe.

Classes Estáticas - Características

- Uma classe estática não pode ser instanciada;
- Classe estática podem ter somente membros estáticos;
- Um Membro estático da classe pode ser acessado pelo próprio nome da classe;
- Uma Classe estática é sealed. Dessa forma uma classe estática não pode ser herdada;
- Uma Classe estática contém apenas os construtores estáticos;
- Uma Classe estática não pode ter construtores de instância;
- Uma Classe estática só pode ser herdada a partir de objetos da classe;

Classes Estáticas - Características

- Uma Classe estática possui a palavra-chave static como modificador;
- Os Construtores estáticos da classe estática são chamados apenas uma vez;
- Uma Classe estática possui somente construtores privados;

Classes Estáticas - Funcionamento

- As informações da classe estática são carregadas pelo **Runtime**, quando o programa que acessa ou faz referência a classe estática está sendo carregado.
- O programa que referência a classe estática não tem controle para especificar quando a classe estática deve ser carregado pelo **Runtime**.
- A classe estática permanece na memória pelo período de vida do domínio de aplicação no qual os programas residem. Ou seja, enquanto o programa está rodando, ela permanece em memória.

Classes Estáticas - Funcionamento

- Como podemos ver **Static** é um recurso disponível na linguagem de programação e sua utilização quando bem feita é válida.
- Mas tenha em mente que usar o recurso **static** de forma indiscriminada, além de violar o paradigma a orientação a objetos, pode te dar muita dor de cabeça.
- Um método estático indica que na verdade não sabe a quem pertence, como ele não é um método de instância, ele não precisa da instância da classe para ser usado.
- Então, a rigor, ele fica meio à vontade usando a assinatura da classe e não usa o estado interno da classe.

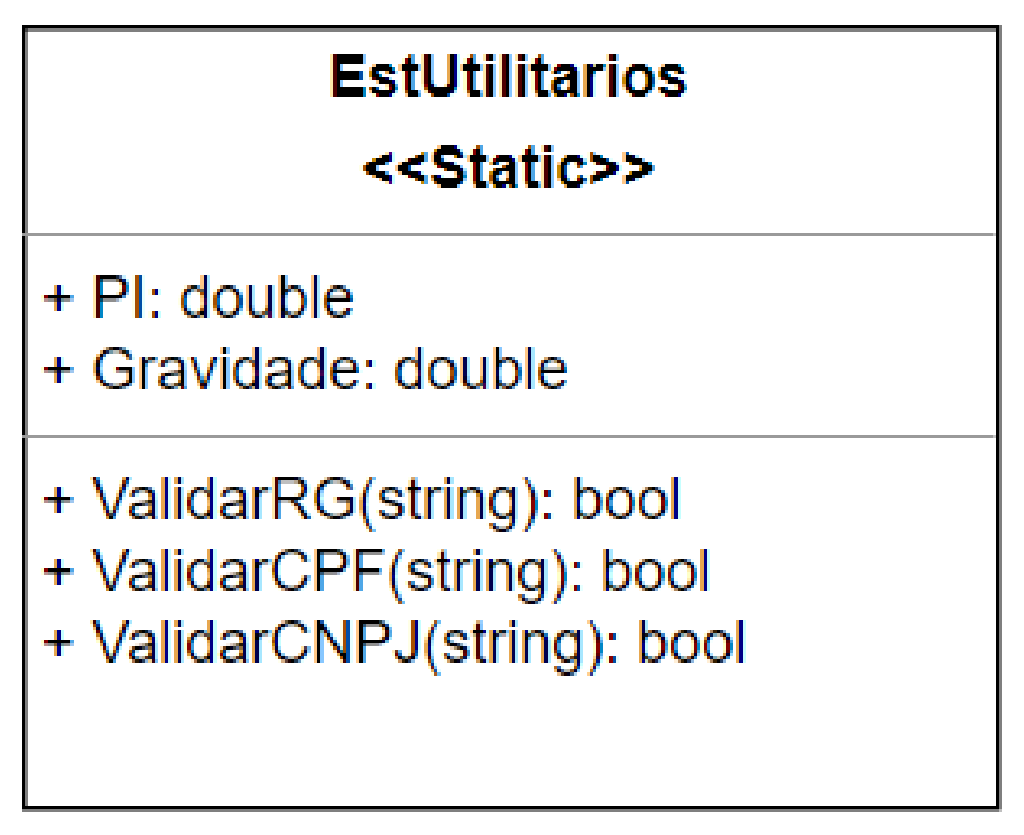
Classes Estáticas - Funcionamento

- Conforme o princípio da responsabilidade (**SRP**), que dita que uma classe deve ter uma única responsabilidade, um método estático estaria violando este princípio, pois ele tende a ter uma responsabilidade que não é a mesma da classe à qual está vinculado.
- Outro problema é que essencialmente um método estático pode ser chamado por qualquer um, a partir de qualquer lugar da sua aplicação.
- Além disso, o método estático pretende ser parte de uma classe, quando na verdade a classe é usada pelo método estático como um ponto de apoio.
- Com base nisso os métodos estáticos estão mais voltados para a programação procedural do que para a orientação a objetos.

Classes Estáticas - Utilização

- Declaração de constantes globais como o número PI. Observe que nos referimos a constantes globais (utilizadas em todo o programa), e não variáveis globais;
- Criação de objetos - principalmente em métodos sobrecarregados, onde podemos usar um construtor estático para ficar mais claro como o objeto está sendo criado.

Classes Estáticas - Representação



- As Boas Práticas recomendam o uso do prefixo ***Est***, em situações comuns. Deve-se levar em conta a camada na qual a classe existe.

LOO

Encontro 04

Prática

Aula Prática

- Otimizando e criando as classes asbtratas:
  - BaseIdentificador
  - BaseDadosComuns
  - BasePessoa
  - BasePessoaFisica
  - BasePessoaJuridica
  - BaseFuncionario
- Otimizando e criando as classes reais:
  - Aluno
  - Fornecedor
  - Professor
  - Tecnico

LOO

Encontro 04

Atividade de Aula

Atividade de Aula

1. Responder a atividade, denominada “LOO – Atividade 04.pdf”, disponível apenas no Github e no Google Drive. VALE NOTA!!!

2. Enviar as respostas justificadas no e-mail do professor, em formato PDF, com o seguinte assunto:

[LOO] ATIVIDADE 04

3. Data de entrega: 04/09/2023, até às 23h00.

Encerramento

Dúvidas e sugestões, entre em contato pelo whatsapp da disciplina, ou mande um e-mail para luiz.a.rodrigues@cogna.com.br