UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE PROCESSAMENTO DE ENERGIA ELÉTRICA

Introdução

Curso de Engenharia de Controle e Automação

DPEE1090 - Programação orientada a objetos para automação

Prof. Renan Duarte

1º semestre de 2024

Introdução

Sobre o professor

Renan Duarte

- Graduação, mestrado e doutorado na UFSM (engenharia elétrica)
- Membro do GEDRE desde 2010
- Emprego na indústria entre 2020 e 2023 (desenvolvimento de hardware e firmware)
- Professor da UFSM desde setembro 2023

Contato

- renan.duarte@gedre.ufsm
- renan.duarte@ufsm.br

gedre inteligência em iluminação

Introdução

Sobre a disciplina

"Programação usando o paradigma da orientação a objetos. Objetos e classes. Abstração e encapsulamento. Herança e polimorfismo. Sobrecarga. Associação. Agregação. Composição. Dependência."

Introdução

Bibliografia

Poucos livros focam **especificamente** na POO e não em alguma linguagem

- COAD, P. Object-oriented programming. Englewood Cliffs, New Jersey: Yourdon Press, 1993. → **Biblioteca CT**
- CARVALHO, Thiago Leite. Orientação a Objetos - Aprenda seus conceitos e suas aplicabilidades de forma efetiva. Casa do Código, 2016. → **Alura**

- https://dotnet.microsoft.com/pt-br/learn/csharp
- Udemy/Alura → **Luiz Otávio Miranda**
- https://josenaldo.github.io/aprendendo-git-e-github/

Introdução

Avaliação

Duas avaliações ao longo do semestre

$$\text{Média Final} = (P1 + P2)/2$$

Avaliação 1: 02/05/2024

Avaliação 2: 27/06/2024

Para aqueles que não atingirem a média mínima de 7,00 e que **mantiverem a frequência mínima necessária (75%)**, será oferecida a oportunidade de realizar um exame de recuperação

Introdução

Avaliação - Exemplo

Fazer um programa para ler os dados de N contribuintes (N fornecido pelo usuário), os quais podem ser pessoa física ou pessoa jurídica, e depois mostrar o valor do imposto pago por cada um, bem como o total de imposto arrecadado.

Os dados de pessoa física são: nome, renda anual e gastos com saúde. Os dados de pessoa jurídica são nome, renda anual e número de funcionários.

As regras para cálculo de imposto são as seguintes:

Pessoa física: pessoas cuja renda foi abaixo de 20000.00 pagam 15% de imposto. Pessoas com renda de 20000.00 em diante pagam 25% de imposto. Se a pessoa teve gastos com saúde, 50% destes gastos são abatidos no imposto.

Exemplo: uma pessoa cuja renda foi 50000.00 e teve 2000.00 em gastos com saúde, o imposto fica: (50000 * 25%) - (2000 * 50%) = 11500.00

Pessoa jurídica: pessoas jurídicas pagam 16% de imposto. Porém, se a empresa possuir mais de 10 funcionários, ela paga 14% de imposto.

Exemplo: uma empresa cuja renda foi 400000.00 e possui 25 funcionários, o imposto fica: 400000 * 14% = 56000.00

Introdução

Objetivos

- Conhecer as principais técnicas de programação orientada a objetos
- Compreender formas de aplicação destas técnicas por meio do desenvolvimento de projetos de software

Introdução

Paradigmas de programação

Programação procedural

- Lista de instruções para informar ao computador o que fazer
- Surgiu na forma de linguagem de máquina
- Fortran foi a primeira linguagem de alto nível a ganhar ampla aceitação
- Dominou o mercado até o surgimento do paradigma orientado a objetos
- Ainda é bastante utilizado, especialmente em sistemas embarcados

Exemplos: Ada, ALGOL, Basic, C, PHP, Java, Cobol, Fortran, Pascal, Python, Lua, Mathematica

Introdução

Paradigmas de programação

Programação orientada a objetos

- Baseado na abstração digital do mundo real, através da composição e interação entre diversas unidades chamadas de objetos e as classes
- Começou a ganhar forma na década de 1960. Contudo, aumentou consideravelmente sua participação no mercado durante o final da década de 1970 e 1980

Exemplos: C++, C#, VB.NET, Java, Object Pascal, Objective-C, Python, SuperCollider, Ruby,Smalltalk, ActionScript, ColdFusion, Javascript, PHP (a partir da versão 4.0), Perl (a partir da versão 5), Visual Basic (a partir da versão 4), VimL

Introdução

Paradigmas de programação

Programação funcional

- Programas são construídos aplicando e compondo funções. É um paradigma de programação declarativo no qual as definições de função são árvores de expressões que mapeiam valores para outros valores, em vez de uma sequência de instruções imperativas que atualizam o estado de execução do programa
- Origens na matemática (cálculo lambda)
- Primeira linguagem de alto nível foi a Lisp (final dos anos 1950)

Exemplos: Common Lisp, Scheme, Clojure, Wolfram Language, Racket, Erlang, Elixir, OCaml, Haskell, F#, C++, C#, Kotlin, Perl, PHP, Python, Go, Rust, Raku, Scala, Java (since Java 8)

Introdução

Paradigmas de programação

Além destes, inúmeros outros paradigmas existem atualmente:

- Programação estruturada
- Programação de passagem de mensagens
- Programação orientada a fluxos
- Programação escalar
- Programação vetorial
- Programação restritiva
- Programação orientada a aspecto
- Programação orientada a regras
- Programação orientada a tabelas
- Programação orientada a fluxo de dados
- Programação orientada a políticas
- Programação orientada a testes
- Programação Genérica

Introdução

Paradigmas de programação

Outros paradigmas (**NÃO RECOMENDADOS** mas extremamente comuns em aplicações do mundo real):

- POG → https://josenaldo.github.io/palestra-pog/#/

  Programação orientada à gambiarra

  Paradigma de desenvolvimento de software no qual se utiliza um improviso planejado temporário permanente na resolução de um problema

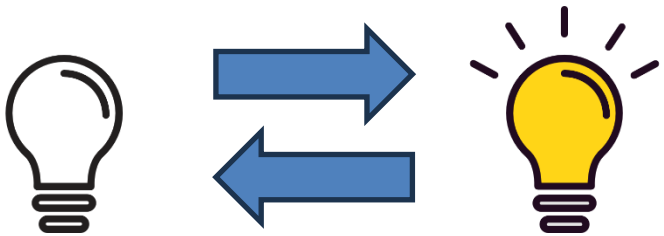
- XGH → https://gohorse.com.br/extreme-go-horse-xgh/

  Extreme Go-Horse

  Não é um paradigma de programação mas sim uma metodologia de desenvolvimento de software

Introdução

Paradigma procedural vs Orientado a objetos



- Variáveis primitivas
- Estruturas
- Funções

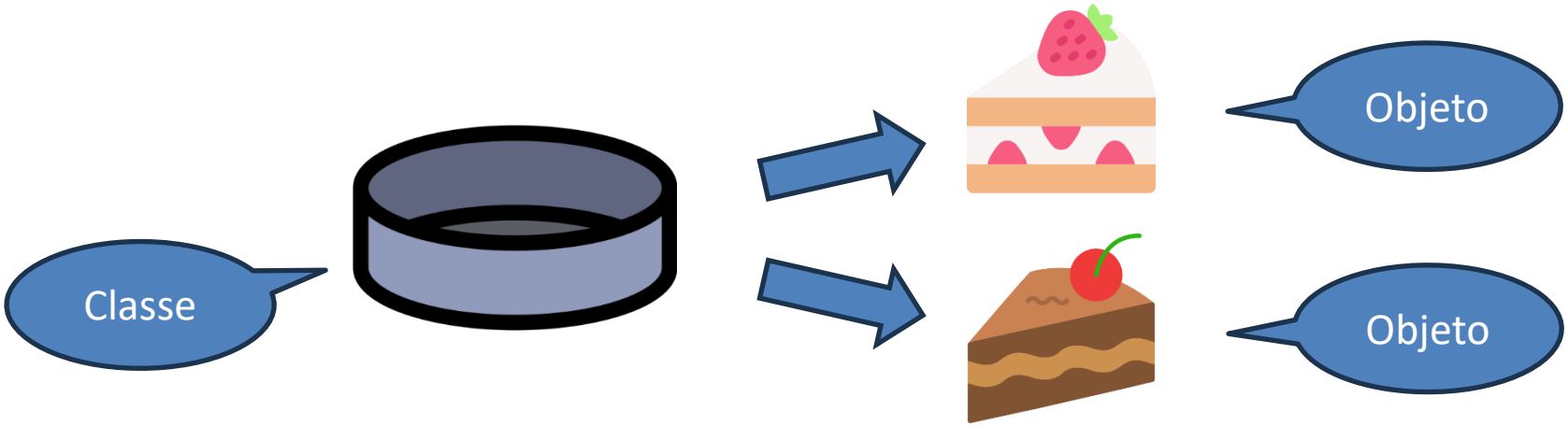
POO modela o mundo real através de objetos

Encapsula dados e comportamento nas chamadas classes

Conceitos básicos da POO

Classes e Objetos

- Classes são modelos para criação de objetos
- Objetos são instâncias concretas de uma classe



Atributos

- Propriedades ou características de um objeto → Dados

Métodos

- Comportamentos ou ações de um objeto → Funções

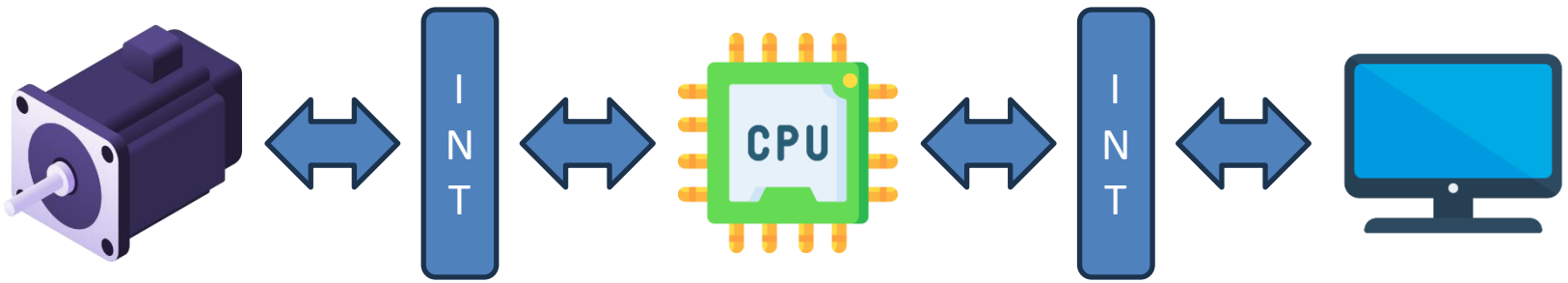
Pilares da POO

Abstração

- Exibir apenas o essencial para o mundo externo
- Permite que se trabalhe de forma incremental → Desacoplamento
- Desenvolvimento de software colaborativo

Interface e implementação

- Interface é a forma como seções do código interagem → Métodos
- Implementação se refere a como estes métodos são escritos



Pilares da POO

Encapsulamento/*data hiding*

- Permite o controle de acesso aos dados
- Cada objeto controla seu próprio estado
- Evita que o programa assuma estados imprevistos

Modificadores

- Public, Private e Protected

Getters e setters

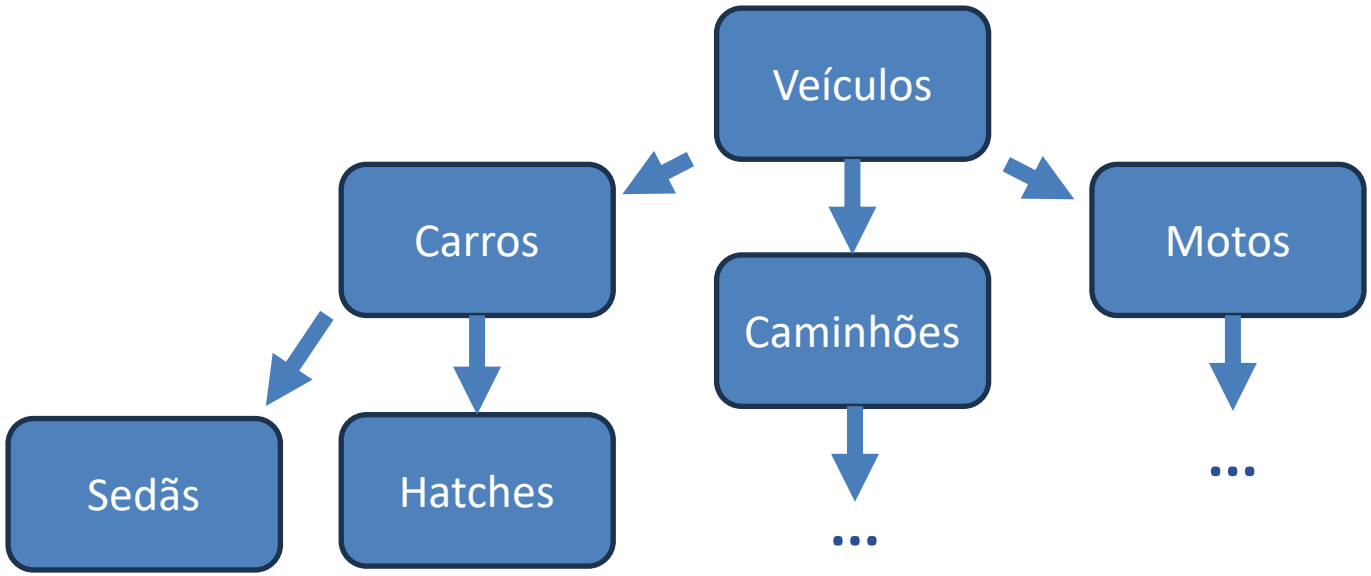
- Em geral, evita-se que uma classe acesse diretamente dados de outra classe → Métodos
- Validação de dados

Pilares da POO

Herança

- Promove a reutilização de código → Herança
- Permite a extensibilidade do programa → Extensão
- Permite estabelecer hierarquias de código → Modificadores

Classe pai/superclasse e classes filhas/subclasses



Pilares da POO

Polimorfismo

- Permite que métodos tenham diferentes formas
- Proporciona reutilização do código
- Permite que uma mesma mesma variável guarde valores de diferentes tipos

Dinâmico e estático

- Dinâmico ocorre durante execução (runtime) → Sobrescrita

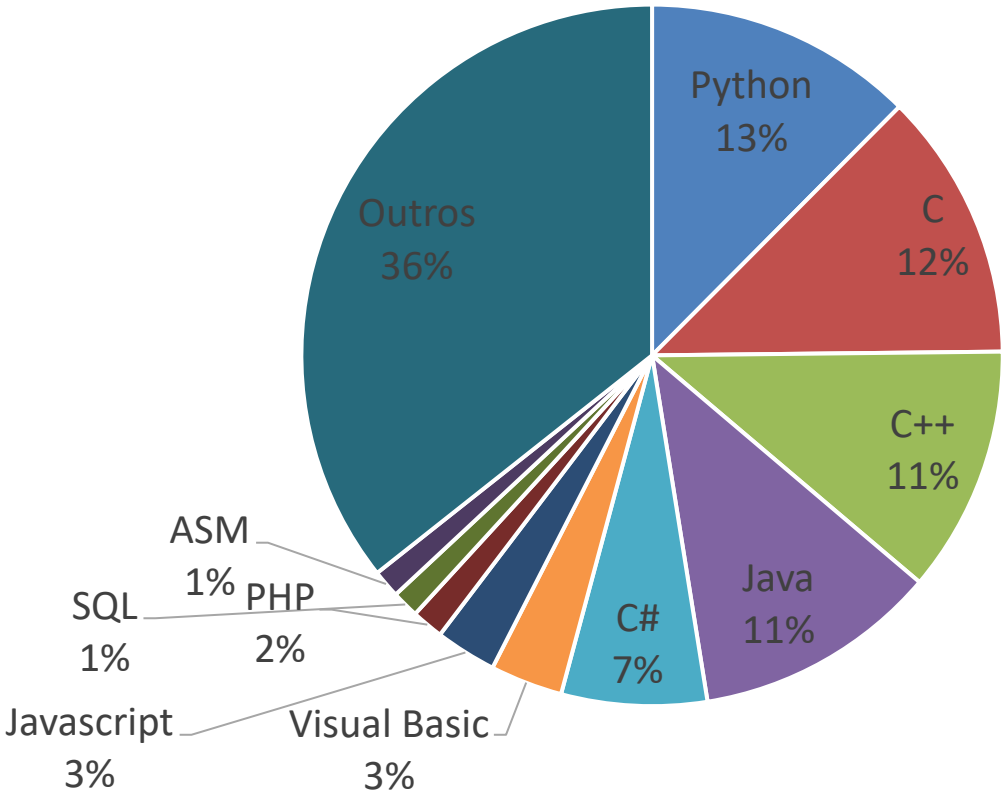
  Subclasses implementam métodos próprios
- Estático ocorre durante a compilação → Sobrecarga

  Métodos com diferentes tipos, quantidades ou ordem dos parâmetros

Linguagens de programação

Linguagens mais utilizadas

Grande maioria é orientada ao objeto ou multiparadigma



TIOBE Index for June 2023

https://www.youtube.com/watch?v=qQXXI5QFUfw

Linguagens de programação

Neste curso – C#

- Desenvolvida pela Microsoft, foi lançada em 2000 como parte da plataforma .NET.
- Sintaxe semelhante a C e C++, facilitando a migração
- Fortemente orientada a objetos
- Atualmente é multiplataforma, permitindo o desenvolvimento de aplicações para diversos sistemas como Windows, Linux e MacOS

Utilizaremos o VisualStudio Community 2022 e a plataforma .NET

- Recursos como compilação, debug e teste
- Suporte em português

Revisão

Próxima aula

Revisão lógica programação usando C#

- Introdução à linguagem de programação C# (.NET, *assembly*, *namespace*, etc.)
- Tipos de dados (tipo valor e tipo referência)
- Conversão implícita e *casting* de tipos de dados
- Convenções de nomenclatura em C#

Dúvidas?

renan.duarte@gedre.ufsm.br

GEDRE – Prédio 10 – CTLAB