

IFRN

POO EM PYTHON

Introdução à Programação Orientada a Objetos

Prof. Gilbert Azevedo

Objetivos

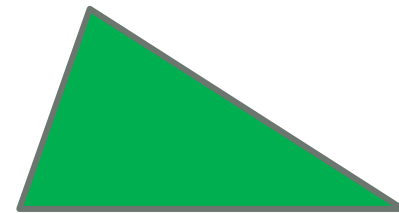
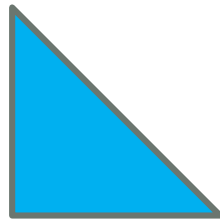
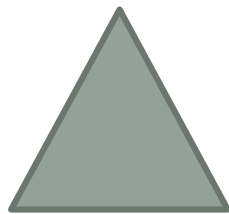
- Entender os conceitos de Entidade, Classificação e Abstração
- Comparar modelos da Programação Estruturada e POO
- Modelar um triângulo nas linguagens C e Python
- Utilizar referências e instâncias de classes

Entidades

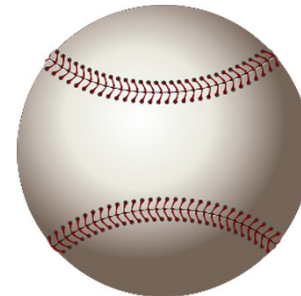
- Entidades são coisas do mundo real que precisam ser representadas em um sistema computacional (*software*)
 - Sistema Comercial: Clientes, Produtos, Vendas, ...
 - Sistema Bancário: Clientes, Contas-corrente, Empréstimos, ...
 - Sistema Acadêmico: Alunos, Professores, Turmas, Cursos, ...
- Na modelagem do sistema, são identificadas suas entidades, as informações que elas armazenam e as operações que elas realizam
 - Entidade: Conta-corrente
 - Dados: cliente, saldo e número da conta.
 - Operações: debitar, creditar, verificar o saldo

Classificação

- Classificação é ação de reunir em grupos objetos que possuam características semelhantes
- Triângulos



- Bolas

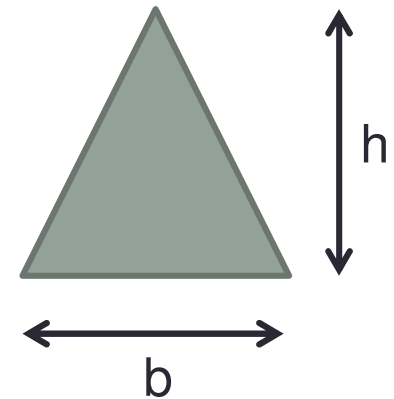


Modelo de Entidades

- Após classificar os objetos e nomear as entidades, é necessário definir um modelo computacional
- Programação Estruturada
 - Estruturas e funções são usadas no modelo das entidades. As estruturas armazenam as informações dos objetos e as funções realizam as ações
- Na Programação Orientada a Objetos
 - Classes são utilizadas para modelar as entidades possuindo campos e métodos.
 - O armazenamento de informações é realizado nos campos ou atributos
 - A realização de operações é feita com métodos

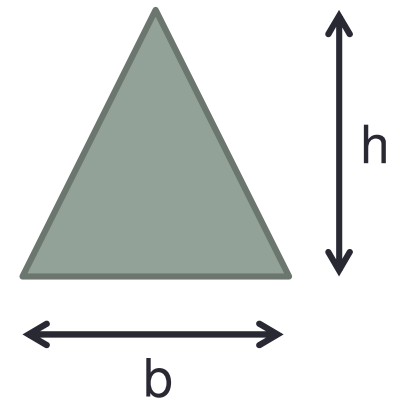
Abstração

- Abstração é a identificação do que é importante na representação de um objeto para o sistema computacional.
- Modelagem de um Triângulo
 - Aplicativo para calcular a área de um triângulo
 - Entidade: Triângulo
 - Dados necessários: base (b) e altura (h) de um objeto triângulo
 - Operações: calcular a área de acordo com a sua base e sua altura
 - Não é necessário, por exemplo, saber a cor do triângulo nem quanto mede cada lado para calcular sua área, portanto, esses dados são irrelevantes para o modelo.



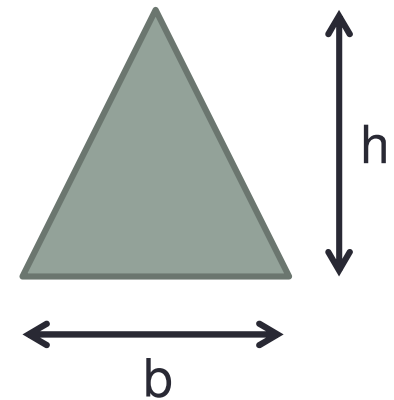
Modelando um Triângulo em C

- Um triângulo em linguagem C (Linguagem Estruturada)
- Modelando os dados (uma estrutura)
 - `struct Triangulo {`
 - `double b, h;`
 - `};`
- Modelando as operações (uma função)
 - `double CalcArea(Triangulo t) {`
 - `return t.b * t.h / 2;`
 - `}`



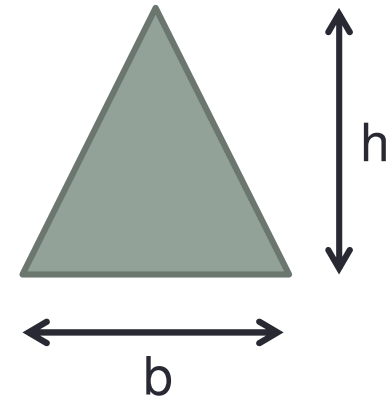
Usando o Triângulo em C

- Usando um triângulo em linguagem C
 - A variável x representa um objeto triângulo com base=10 e altura=20
 - `int main() {`
 - `Triangulo x;`
 - `x.b = 10;`
 - `x.h = 20;`
 - `printf("%lf", CalcArea(x));`
 - `}`



Limitações do Modelo

- Limitações da Programação Estruturada
 - Dados e as operações não ficam agrupados em um tipo de variável
 - O acesso direto aos campos do objeto possibilita a atribuição de valores errados, levando o objeto a um estado inválido
- ```
int main() {
 Triangulo x;
 x.b = -10;
 x.h = -20;
}
```
- Não existe, no mundo real, figuras com dimensões negativas



# Modelando um Triângulo em Python

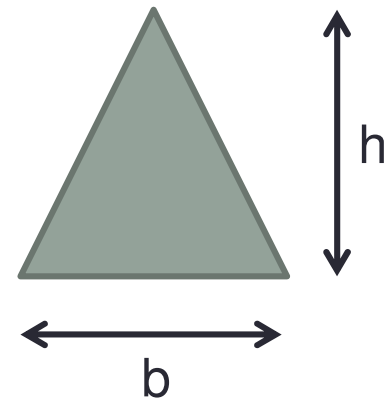
- Na POO, as entidades são modeladas em uma classe:
  - Os dados são os campos (ou atributos) da classe
  - As operações são realizadas pelos métodos
  - A classe com seus campos e métodos define um novo tipo de variável
- `class Triangulo:` # Tipo de variável
- `def __init__(self):`
- `self.b = 0` # Campos ou atributos
- `self.h = 0`
- `def calc_area(self):` # Método
- `return self.b * self.h / 2`

# Referência e Instância

- No Python, as variáveis de uma classe são alocadas quando uma atribuição com o nome da classe é realizada
  - `x = Triangulo()`
  - A variável `x` é chamada de referência
  - A instância da classe (objeto) é retornada pela instrução `Triangulo()`
- O local do objeto na memória é apresentado ao usar a função *print*
  - `print(x)`
  - `#<__main__.Triangulo object at 0x00000227987FADA0>`

# Usando o Triângulo em Python

- Usando um triângulo em linguagem Python
  - A variável x é uma referência para um objeto triângulo com base=10 e altura=20. Campos e métodos são chamados com a referência.
  - `x = Triangulo()`
  - `x.b = 10` # altera a base para 10
  - `x.h = 20` # altera a altura para 20
  - `print(x.calc_area())` # calcula a área



# Passo a Passo

- Utilizando a classe Triangulo para calcular a área de um triângulo
- 1º passo. Instancia a classe: Referência
  - `x = Triangulo()`
  - `x` recebe a instancia criada com base e altura iguais a zero
  - `x` é uma referência para o objeto alocado com acesso aos atributos e métodos

# Passo a Passo

- 2º passo: Ajustar as dimensões do triângulo, utilizando os campos da classe
  - `x.b = 10`
  - `x.h = 20`
  - O objeto agora tem base=10 e altura=20. Nesse ponto, ainda é possível atribuir valores negativos para os campos. Isso será ajustado no futuro.
- 3º passo: Calcular a área, utilizando o método para esta operação
  - `x.calc_area()`
  - O cálculo da área é realizado.

# Referências

- Classes em Python
  - <https://docs.python.org/3/tutorial/classes.html>
- Pixabay
  - Football Image by [OpenClipart-Vectors](#) from [Pixabay](#)
  - Basketball Image by [OpenClipart-Vectors](#) from [Pixabay](#)
  - Baseball Image by [OpenClipart-Vectors](#) from [Pixabay](#)