

>>> Programação Orientada a Objetos (POO)

... Aplicação - Geometria Vetorial

Prof: André de Freitas Smaira

>>> Exemplo

- * Vamos fazer um exemplo **mais completo**
- * Apesar de ser exemplo haverá **detalhes novos** que podem ajudar
- * Uma biblioteca de **geometria analítica vetorial 2D** usando **classes**
- * **Vantagens** da geometria analítica vetorial? Alguns exemplos:
 - * Reta vertical
 - * Cruzamentos

>>> Métodos Dunder

* **Lembrando:** métodos no formato **operator<operador>** que determinam o funcionamento de operações básicas, como **+**, **-**, *****, **/**, **%**, **<<**, etc.

operator+
operator-
operator*
operator/
operator%
operator++
operator--
operator==
operator!=
operator<
operator>

operator<=
operator>=
operator&&
operator||
operator!
operator&
operator|
operator^
operator~
operator<<
operator>>

operator=
operator+=
operator-=
operator*=
operator/=
operator%=
operator&=
operator|=
operator^=
operator<<=esquerda
operator>>=

operator[]
operator()
operator->
operator new
operator new[]
operator delete
operator delete[]
operator type

>>> Exemplo

- * Essa aula vai ser **inteiramente prática**, então daqui pra frente vamos apenas **programar**...
- * E vou **"desenhar" um pouco** para explicar os métodos mais complicadas

```
>>> Exemplo
```

```
Vamos implementar...
```

```
* classe Ponto
    * Ponto
    * operator[]
    * operator<<
    * operator==
    * operator<
    * distancia
```

```
>>> Exemplo
```

```
Vamos implementar...
```

```
* classe Vetor que herda de Ponto
```

```
    * Vetor
```

```
    * operator+
```

```
    * operator-
```

```
    * operator*
```

```
    * operator^
```

```
    * abs
```

```
    * operator/
```

```
    * operator||
```

```
    * operator<<
```

```
    * operator>>
```

```
    * versor
```

```
>>> Exemplo
```

```
Vamos implementar...
```

```
* classe Reta
```

```
    * Reta
```

```
    * operator==
```

```
    * operator<<
```

```
    * operator[]
```

```
    * operator||
```

```
    * origem
```

```
    * vetor
```

```
    * interseccao
```

```
    * distancia
```

```
    * perpendicular
```

```
    * operator<
```

```
    * mesmo_lado
```

```
>>> Exemplo
```

Vamos implementar...

```
* classe SemiReta que herda de Reta
  * SemiReta (precisa?)
  * operator[]
  * interseccao
  * operator-
  * operator<<
  * distancia
  * perpendicular
  * operator<
```



```
>>> Exemplo
```

Vamos implementar...

```
* classe SegmentoReta que herda de SemiReta
  * SegmentoReta (precisa?)
  * operator[]
  * abs
  * operator<<
  * operator==
  * distancia
  * inicio
  * fim
  * interseccao
```

```
>>> Exemplo
```

```
Vamos implementar...
```

```
* classe Poligono
```

```
    * Poligono
```

```
    * n_lados
```

```
    * operator[]
```

```
    * vertices
```

```
    * perimetro
```

```
    * area
```

```
>>> Exemplo
```

```
Vamos implementar...
```

```
* classe Triangulo que herda de Poligono
```

```
    * Triangulo
```

```
    * retangulo
```

```
    * acutangulo
```

```
    * obtusangulo
```

```
>>> Exemplo
```

Vamos implementar...

- * classe PoligonoRegular que herda de Poligono

- * PoligonoRegular

- * centro

- * angulo_central

- * lado

- * raio_inscrita

- * raio_circunscrita

```
>>> Exemplo
```

```
Vamos implementar...
```

```
* classe Circulo que herda de PoligonoRegular
```

```
    * Circulo
```