Estruturas Básicas

Conceitos que criam as estruturas básicas da OO:

* Classe
* Atributo
* Método
* Objeto
* Mensagem

Classes

"É uma estrutura que abstrai um conjunto de objetos com características similares. Uma classe define o comportamento de seus objetos através de métodos e os estados possíveis destes objetos através de atributos. Em outros termos, uma classe descreve os serviços providos por seus objetos e quais informações eles podem armazenar.”

(<https://pt.wikipedia.org/wiki/Classe_(programação)>)

Exemplos:

* Bola Carro
* Venda Viagem
* Comprador
* Computador

Dicas:

- Substantivos

- Nome significativos

- Contexto deve ser considerado

• Códigos:

Java

class Carro {

}

C#

class Carro {

}

Python

class Carro:

pass

Atributo

"É o elemento de uma classe responsável por definir sua estrutura de dados. O conjunto destes será responsável por representar suas características e fará parte dos objetos criados a partir da classe."

Característica

Exemplos:

Bola • diâmetro

Carro • cor

Venda • valor

Viagem • distancia

Comprador • nome

Computador • memoria

Atributo x Variável

A: O que é próprio e peculiar a alguém ou a alguma coisa.

V: Sujeito a variações ou mudanças; que pode variar; inconstante, instável

• Dicas:

- Substantivos e adjetivos

- Nome significativos

- Contexto deve ser considerado

- Abstração

- Tipos adequados

Códigos:

Java

class Carro {

int portas;

}

C#

class Carro {

int portas;

}

Python

class Carro: portas = 0

Método

"É uma porção de código (sub-rotina) que é disponibilizada pela classe. Este é executado quando é feita uma requisição a ele. Um método serve para identificar quais serviços, ações, que a classe oferece. Eles são responsáveis por definir e realizar um determinado comportamento"

Exemplos:

Carro • ligar

Venda • calcular total

Comprador • realizar troca

Criação:

Java e C#

- Visibilidade

- Retorno = Atrelado ao tipo de dado

- Nome

- Parâmetros = Caso precise

Python

- def

- Nome

- Parâmetro

Dicas:

- Verbos

- Nome significativos

- Contexto deve ser considerado

Códigos:

Java

class Carro {

void frear(){

...

}

}

C#

class Carro { void frear(){

...

}

}

Python

class Carro:

def frear() ..

Dois métodos especiais:

✔ Construtor

Sempre abaixo dos atributos, depois getter e setters e depois

Destrutor.

Java

class Carro {

Carro(){

...

}

}

C#

class Carro {

Carro(){

...

}

}

Python

class Carro:

def \_\_init\_\_(self) :

o Destrutor

Auxilia destruição do objeto

Java

class Carro {

void finalize (){

...

}

}

C#

class Carro {

~Carro(){

...

}

}

Python

class Carro:

def \_\_init\_\_(self) :

Sobrecarga:

o Mudar a assinatura de acordo com a necessidade

– Assinatura: nome + parâmetros

Java

m1()

m1(int i)

m1(float f)

m1(String s, long l)

m1(long l, String s)

C#

M1()

M1(int i)

M1(float f)

M1(String s, long l)

M1(long l, String s)

Python

o Porque usar?

Método

Main inicia e termina aplicação