



www.python.pro.br

Polimorfismo

Conceito

Interface

Operação

Classe Abstrata

Calculadora

Duck Typing

Conclusão

Tarefa *

Conceito

Várias Formas

Tarefa pode ser feita de várias maneiras

Define-se um contrato com "O que" deve ser feito

Implementa-se o contrato a posteriori

Diminui grau de dependência *

Interface

Não existe no Python

Define formalmente o contrato

Exemplos Mundo Real: Direção, Tomada

Classes concretas implementam interface*

Operação

Responsabilidade: efetuar calculo

Palavra reservada interface

Métodos sem corpo

Palavra reservada implements

Classes Adicao e Subtracao *

Classe Abstrata

Pode definir métodos abstratos, como a interface

Pode definir métodos concretos

Impossível de instanciar *

Calculadora

Classe responsável por conter operações

Realiza cálculos baseado na escolha

Classes concretas definem como obter dados do usuário

Palavra reservada `abstract`

Classes concretas `CalculadoraPrefixa` e `CalculadoraInfixa` *

Duck Typing

Contrato Informal

Se voa como Pato, nada como Pato, então é um pato

Não existe no Java

Fazer problema da Calculadora com Herança e Duck typing no Python*

Conclusão

Polimorfismo em Java apenas com Herança

Python tem Duck Typing

Dar preferência a Interfaces a classes abstratas para não perder única oportunidade de se fazer herança no Java *

Tarefa

Criar classe Manada, que irá possuir Animais

Criar método de adição de animais

Criar Classe ManadaVirgula

Criar Classe ManadaSustenido

Gerar String Respectivas

Utilizar classes Gato e Cachorro criadas anteriormente *

Obrigado

renzo@python.pro.br
@renzoprobr

