PROGRAMAÇÃO ORIENTADA A OBJETOS - POO

Sistemas para internet

Prof.º: Paulo de Tarso Oliveira da Silva

Contatos:

(96) 99142-4429

05.paulotarso@gmail.com

FACULDADE META

Polimorfismo - *Interfaces*

Aula 5

FACULDADE META

Interfaces

As ***interfaces*** são padrões definidos através de **contratos** ou especificações.

Um contrato define um determinado conjunto de métodos que serão implementados nas classes que assinarem esse contrato. Uma interface **é 100% abstrata**, ou seja, os seus métodos são definidos como *abstract*, e as variáveis por padrão são sempre constantes.

Uma interface é definida através da palavra reservada "*interface*". Para uma classe implementar uma *interface* é usada a palavra "***implements***".

Interfaces

Como a linguagem PHP não tem herança múltipla, as *interfaces* ajudam nessa questão, pois bem se sabe que uma **classe** pode ser herdada apenas uma vez, mas pode implementar inúmeras interfaces.

As classes que forem implementar uma interface terão de adicionar **todos os métodos** da interface ou se transformar em uma classe abstrata.

Criando uma *Interface*

Criando uma Interface


```php
<?php

interface Conta
{
    public function depositar($valor);

    public function sacar($valor);

    public function getSaldo();
}
```

Implementando *Interface* em **ContaCorrente**


```php
<?php

class ContaCorrente implements Conta
{
    protected $saldo = 0;
    protected $taxaOperacao = 0.45;

    public function depositar($valor)
    {
        $this->saldo += $valor - $this->taxaOperacao;
    }

    public function sacar($valor)
    {
        $this->saldo -= $valor + $this->taxaOperacao;
    }

    public function getSaldo()
    {
        return $this->saldo;
    }
}
```

FACULDADE META

Implementando *Interface* em **ContaPoupanca**


```php
<?php

class ContaPoupanca implements Conta
{
    protected $saldo = 0;

    public function depositar($valor)
    {
        $this->saldo += $valor;
    }

    public function sacar($valor)
    {
        $this->saldo -= $valor;
    }

    public function getSaldo()
    {
        return $this->saldo;
    }
}
```

Classe **GeradorExtratos**

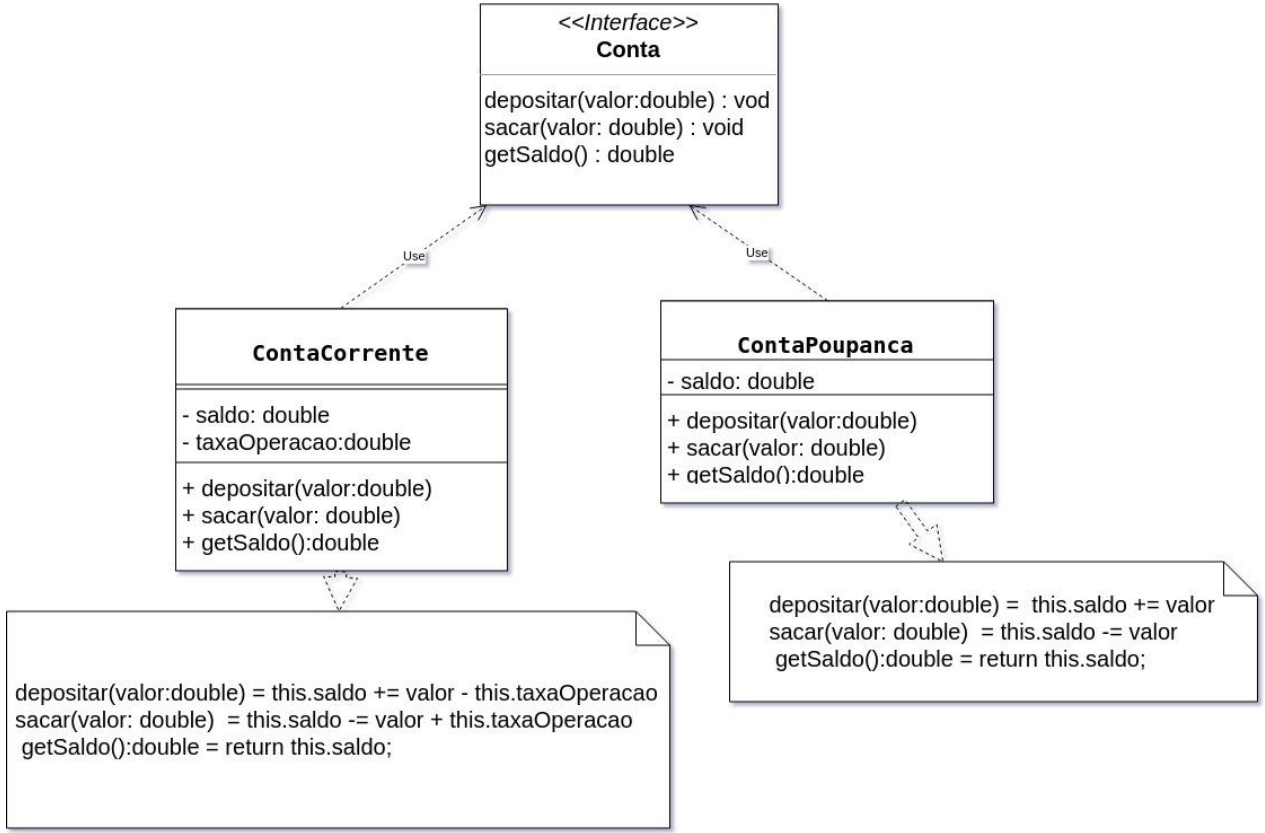
Podemos Criar uma classe com a seguinte estrutura:

```php
<?php

/**
*
*/
class GeradorDeExtrato
{
    public function gerarExtrato(Conta $conta)
    {
        "Saldo atual: " $conta->getSaldo();
    }
}
```

O método “**gerarExtrato**”, mostra a entrada de um parâmetro do tipo **Conta**, assim esse parâmetro poderá receber uma instância de qualquer classe que Implemente a Interface Conta;

Exemplo em diagrama



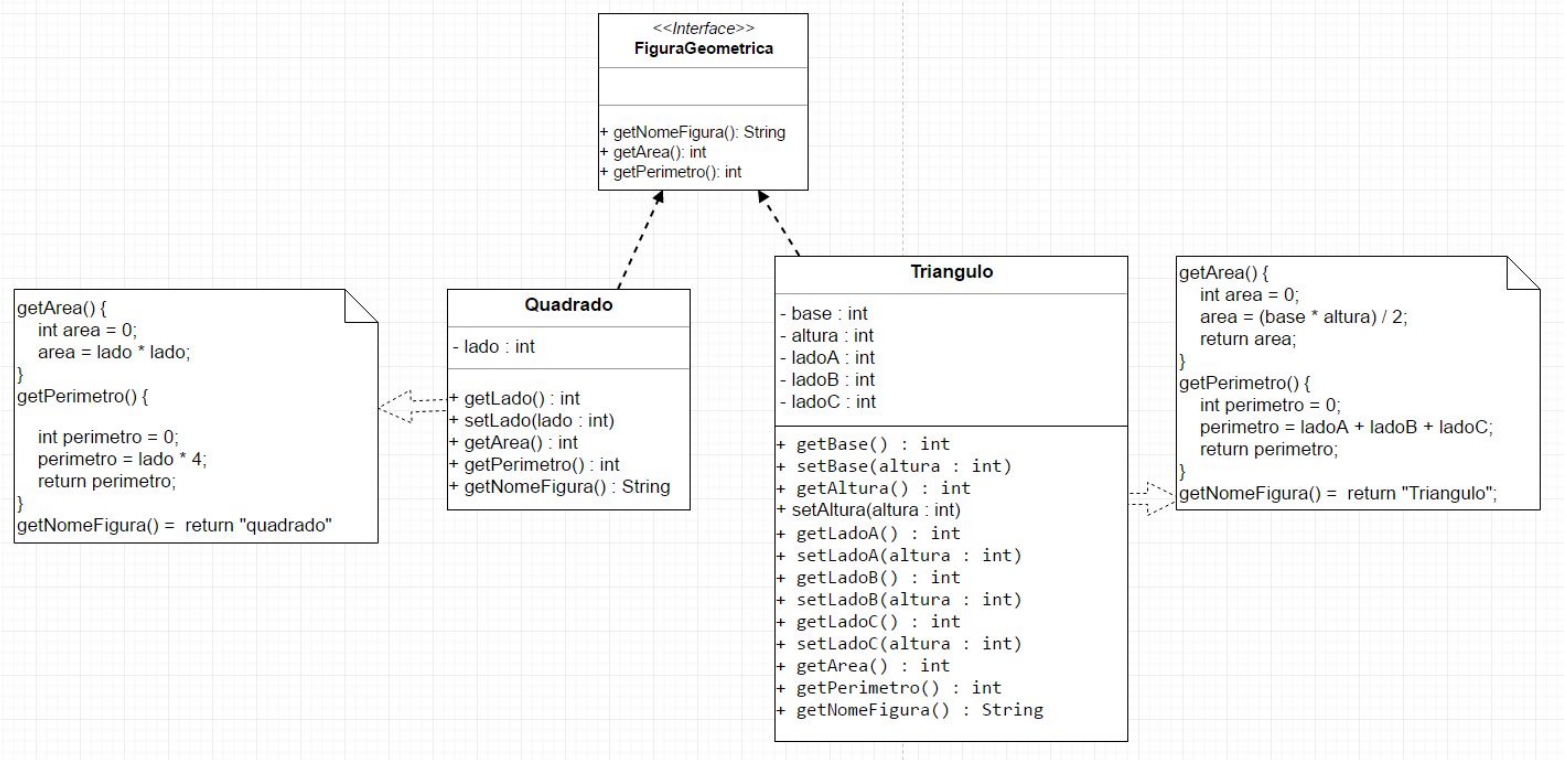
Vantagens de utilização de *Interfaces*

Vantagens de utilização de *Interfaces*

Como é possível ver, ambas as classes seguiram o contrato da *interface Conta*, porém cada uma delas a implementou de maneira diferente.

Ao contrário da herança que limita uma classe a herdar somente uma classe pai por vez, é possível que uma classe implemente várias *interfaces* ao mesmo tempo.

Exercício - Crie a seguinte estrutura;



Obrigado!

Referências Bibliográficas.

MANZANO, José Augusto G., COSTA JR., Roberto da. **Programação de Computadores com Java**. Érica, 2014.

FURGERI, Sérgio. **Java 8 - Ensino Didático - Desenvolvimento e Implementação de Aplicações**. Érica, 2015.