

```
<?php
```

```
//CLASSES & MÉTODOS ABSTRATOS:
```

```
//Conceito:
```

//Só para relembrar, classes e métodos abstratos são aquelas classes e métodos criados com o único objetivo de serem herdados por classes filhas e terem métodos que sofram polimorfismo por parte das classes filhas.

//No PHP nós definimos uma classe abstrata usando a palavra reservada "abstract", temos usar essa palavra antes da formulação da classe e nos seus respectivos métodos, se usarmos a palavra reservada "abstract" somente nos métodos e não nas classes o PHP apresentará erro.

//Vejam os um exemplo onde temos uma classe mãe Conta que irá herdar seus métodos sobre as classes ContaCorrente e ContaPoupanca, onde haverá um método de tarifa que deverá ser implementado de forma diferente entre cada uma dessas classes herdeiras...

```
abstract class Conta {
```

```
    protected $nome;
```

```
    protected $documento;
```

```
    protected function __construct(string $nome, string $documento){
```

```
        $this->nome = $nome;
```

```
        $this->documento = $documento;
```

```
    }
```

```
    abstract protected function tarifa(); //Detalhe importante, os métodos abstratos não podem ter corpo, por que o
```

```
        //objetivo é que o corpo delas seja implementado nas classes filhas,
```

```
    } //então o PHP não permite a implementação de corpo nelas, ou seja, as
```

```
        //chaves "{}"
```

```
class ContaCorrente extends Conta { //No momento que você implementar essa classe o PHP irá mostrar uma mensagem de
```

```
        //erro, essa mensagem é só para alertá-lo de que um método abstrato existente
```

```
    private $saldo;
```

```
        //nessa classe também deverá ser implementado na classe filha...
```

```
    private $valorTarifa = 1.05;
```

```

public function __construct(string $nome, string $documento, float $saldo){

    parent::__construct($nome, $documento);
    $this->saldo = $saldo;

}

public function tarifa():float //Somos obrigados a implementar um método abstrato sobre a nossa classe filha.
{
    return $this->saldo - $this->valorTarifa;
}

public function verExtrato()
{
    echo "O saldo de $this->nome é de: R$: " . $this->tarifa() . PHP_EOL . PHP_EOL;
}

}

class ContaPoupanca extends Conta {

    private $saldo;
    private $valorTarifa = 0.55; //Note que o valor de tarifa é outro...

    public function __construct(string $nome, string $documento, float $saldo){

        parent::__construct($nome, $documento);
        $this->saldo = $saldo;

    }

    public function tarifa():float //Veja que temos o mesmo método herdado da classe abstrata mas o resultado será
    {
        //diferente por causa do valor de tarifa...
        return $this->saldo - $this->valorTarifa;
    }
}

```

```

    }

    public function verExtrato()
    {
        echo "O saldo de $this->nome é de: R$: " . $this->tarifa() . PHP_EOL . PHP_EOL;
    }
}

$fabiano = new ContaCorrente('Fabiano', '124.512.634-1', 1500); //Note que Fabiano e Roberto possuem o mesmo valor
$fabiano->verExtrato();                                           //de saldo, mas como possuem tipos de contas
                                                                //diferentes o valor de suas tarifas será diferente
$roberto = new ContaPoupanca('Roberto', '452.478.634-1', 1500); //trazendo um saldo maior e outro menor...
$roberto->verExtrato();

```

?>

[Running] php "c:\Users\Almoxarifado\Documents\php\arquivos_das_aulas\39-Classes_&_Metodos_Abstratos.php"

O saldo de Fabiano é de: R\$: 1498.95

O saldo de Roberto é de: R\$: 1499.45

[Done] exited with code=0 in 0.076 seconds