

Classes e Objetos

Classe

Uma classe é um conjunto de variáveis e funções (atributos e métodos) relacionadas a essas variáveis (objetos). Uma vantagem da utilização é poder usufruir do recurso de encapsulamento de informação.

Com o encapsulamento, o usuário de uma classe não precisa saber como ela é implementada, bastando para a utilização conhecer a interface, ou seja, as funções disponíveis.

Uma classe é um tipo, portanto não pode ser atribuída a uma variável. Para definir uma classe, deve-se utilizar a seguinte sintaxe:

```
class Nome_da_classe {
    var $variavel1;
    var $variavel2;
    function funcao1 ($parametro) {
        /* === corpo da função === */
    }
}
```

Objeto

Como foi dito anteriormente, classes são tipos, e não podem ser atribuídas a variáveis. Variáveis do tipo de uma classe são chamadas de objetos, e devem ser criadas (instanciadas) utilizando o operador `new`, seguindo o exemplo abaixo:

```
$variavel = new $nome_da_classe;
```

Para utilizar as funções definidas na classe, deve ser utilizado o operador "`->`", como no exemplo:

```
$variavel->funcao1()
```

A variável `$this`

Na definição de uma classe, pode-se utilizar a variável `$this`, que é o próprio objeto. Assim, quando uma classe é instanciada em um objeto, e uma função desse objeto na definição da classe utiliza a variável `$this`, essa variável significa o objeto que estamos utilizando.

Como exemplo da utilização de classes e objetos, podemos utilizar a classe conta, que define uma conta bancária bastante simples, com funções para ver saldo e fazer um crédito.

```
class conta {
    var $saldo;
    function saldo() {
        return $this->saldo;
    }
    function credito($valor) {
        $this->saldo += $valor;
    }
}

$minhaconta = new conta;
$minhaconta->saldo(); // a variavel interna não foi
// inicializada, e não contém
// valor algum

$minhaconta->credito(50);
$minhaconta->saldo(); // retorna 50
```

SubClasses

Uma classe pode ser uma extensão de outra classe. Isso significa que ela herdará todos os atributos (as variáveis) e métodos (funções) da outra classe, e ainda terá as que forem adicionadas pelo programador.

Em PHP não é permitido utilizar herança múltipla, ou seja, uma classe pode ser extensão de apenas uma outra.

Para criar uma classe estendida, ou derivada de outra, deve ser utilizada a palavra reservada `extends`, como pode ser visto no exemplo seguinte:

```
class novaconta extends conta {
    var $numero;
    function numero() {
        return $this->numero;
    }
}
```

A classe acima é derivada da classe conta, tendo as mesmas funções e variáveis, com a adição da variável `$numero` e a função `numero()`.

Construtores

Um construtor é uma função definida na classe que é automaticamente chamada no momento que a classe é instanciada (através do operador `new`). O

construtor deve ter o nome `__constructor()` após a palavra `function`. Obs. São dois `_` sem espaços antes da palavra `constructor()`

Veja o exemplo:

```
class conta {
  var $saldo;

  function __constructor() {
    $this->saldo = 0;
  }

  function saldo() {
    return $this->saldo;
  }
  function credito($valor) {
    $this->saldo += $valor;
  }
}
```

Podemos perceber que a classe `conta` agora possui um construtor, que inicializa a variável `$saldo` com o valor `0`.

Um construtor pode conter argumentos, que são opcionais, o que torna esta ferramenta mais poderosa. No exemplo acima, o construtor da classe `conta` pode receber como argumento um valor, que seria o valor inicial da conta.

Vale observar que para classes derivadas, o construtor da classe pai não é automaticamente herdado quando o construtor da classe derivada é chamado.

Exercícios

- 1) Crie uma classe chamada **Funcionário** que contenha os argumentos **Nome, telefone e salário**. Faça os **métodos incluir funcionário, apagar funcionário, listar funcionário, aumentar salário, ver valor do salário**.
- 2) Crie uma subclasse a partir da classe anteriormente criada que adicione o **e-mail** ao usuário. Para isso crie um argumento `e-mail` e os métodos para atribuir, apagar e listar o `e-mail` do funcionário.

- 3) Crie uma página com um menu em HTML que aponte páginas a serem escolhidas pelo usuário para que, através de formulários, ele possa atribuir, alterar e visualizar os valores do objeto do exercício 2.