

Noções básicas de orientação a objetos

1. Definições:

- Objeto:** O objeto é um tipo abstrato que contém dados mais os procedimentos que manipulam esses dados.
- Mensagens:** É uma informação enviada ao objeto para que ele se comporte de uma determinada maneira. Um programa orientado a objetos em execução consiste em envios, interpretações e respostas às mensagens.
- Métodos:** São procedimentos residentes nos objetos que determinam como eles irão atuar ao receber as mensagens.
- Variáveis de Instância:** São variáveis que armazenam informações ou dados do próprio objeto. Podem também ser chamadas de propriedades do objeto.
- Classe:** Representa um tipo específico de objeto. Uma classe é composta pela sua descrição, que identifica tanto as variáveis de classe (*propriedades da classe*) quanto os métodos.
- Subclasse:** Uma nova classe originada de uma classe maior (classe-pai).
- Instância:** São os objetos de uma classe. Cada objeto utilizado em uma aplicação pertencente a uma classe é uma instância dessa classe.
- Hereditariedade:** É um mecanismo que permite o compartilhamento de métodos e dados entre classes, subclasses e objetos. A hereditariedade permite a criação de novas classes programando somente as diferenças entre a nova classe e a classe-pai.
- Encapsulamento:** É um mecanismo que permite o acesso aos dados do objeto somente através dos métodos desse objeto. Nenhuma outra parte do programa pode operar sobre os dados de nenhum objeto. A comunicação entre os objetos é feita apenas através de mensagens.

- Polimorfismo:** Uma mesma mensagem pode provocar respostas diferentes quando recebidas por objetos diferentes. Com o polimorfismo, pode-se enviar uma mensagem genérica e deixar a implementação a cargo do objeto receptor da mensagem.
- Persistência:** É a permanência de um objeto na memória. Quando um objeto é necessário, ele é criado na memória (métodos construtores). Quando ele não for mais necessário, é destruído da memória (métodos destrutores). Quando um objeto é destruído, seu espaço na memória é liberado automaticamente. Este processo recebe o nome de “coleta de lixo” (*garbage collection*).
- Ligação dinâmica:** É a junção de todas as componentes de um programa durante a execução do mesmo. A ligação é dita estática quando a junção das componentes é feita durante o processo de compilação (*linkedição*).
- Programação Visual:** É uma forma de se programar que está vinculada à programação orientada a objetos. Consiste basicamente em se alterar os objetos desenhando-os na tela e não alterando seus dados numéricos, embora algumas linguagens visuais operem nas duas formas.

2. Comparação entre orientação a objetos e técnicas tradicionais:

Técnicas Orientadas a objetos	Técnicas tradicionais
Métodos	Procedimentos ou funções
Variáveis de Instância	Dados
Mensagens	Chamadas de procedimentos ou de funções
Classes	Tipos de dados
Hereditariedade	(Não existe técnica similar)
Chamadas sob o controle do sistema	Chamadas sob o controle do programador.

3. Exemplos de criação de objetos em pseudocódigo:

- **Classe:**

```
Classe Empregado: objeto {Declaração genérica de uma classe}
  var
    Nome: string          {Variável de instância}

  público                {Declaração dos métodos}
    Procedimento Adicionar (nome: string, idade: inteiro)
    Procedimento Apagar (nome: string)
    Procedimento Alterar (nome: string, idade: inteiro)
  fim

  privado
    { Aqui são declarados método particulares do }
    { objeto e o corpo dos métodos públicos      }
  fim
fim classe
```

- **Hereditariedade:**

```
Classe EmpregadoFeminino: Empregado {Subclasse do tipo}
                                     {Empregado}

  Var
    NumFilhos: inteiro

  Público
    Procedimento Adicionar (nome: string, idade: inteiro,
    Filhos: inteiro)
    Procedimento Alterar (nome: string, idade: inteiro,
    Filhos: inteiro)
  Fim

  Privado
    {Aqui vão somente as declarações e o corpo dos}
    {métodos específicos de EmpregadoFeminino    }
  Fim
fim classe
```

A declaração acima herda todas as variáveis de instância e métodos da classe Empregado, acrescentando NumFilhos e modificando Adicionar e Alterar.

- **Uso em programas**

Algoritmo TesteObjeto

```
var  
    Emp1,Emp2    : Empregado  
    EmpF1,EmpF2 : EmpregadoFeminino
```

```
inicio  
|  
Emp1.Adicionar("Pafúncio",32)  
Emp2.Adicionar("Joaquim",18)  
EmpF1.Adicionar("Ana",23,2)  
EmpF2.Adicionar("Maria",19,0)  
|  
Emp1.Apagar("Pafúncio")  
EmpF1.Apagar("Ana")  
|  
Emp2.Alterar("Joaquim",23)  
EmpF2.Alterar("Maria",29,0)  
|  
fim
```

4. Exercícios:

- Classe Quadrado: propriedades Lado, Cor. Métodos Atribuir Lado, Atribuir Cor, Desenhar e Apagar.
- Subclasse Retângulo: propriedade Base. Método Atribuir Base.
- Fazer um trecho de programa teste.