

INTRODUÇÃO AOS PARADIGMAS DE PROGRAMAÇÃO

Adejair Batista Vieira Júnior¹

RESUMO

Este artigo busca fazer uma introdução sobre os paradigmas de programação, apresentando alguns métodos, descrevendo a estruturação de alguns dos principais paradigmas de programação e quais as respectivas linguagens de programação detêm de alguns dos paradigmas citados.

Palavras-chave: Paradigmas de programação 1. Linguagens de Programação 2. Recursão 3.

¹ Graduando em Engenharia de Software pela Universidade de Rio Verde, Rio Verde, GO.

² João Dionísio Paraíba, mestre em Ciência da Computação.

1 INTRODUÇÃO

Paradigmas de computação são estilos de códigos suportados por linguagens de programação que detêm de características comuns (TUCKER, 2001). Paradigmas de programação podem ser considerados como uma forma de pensar ao desenvolver programas de computadores seguindo uma determinada estruturação.

Um fator que influenciou a criação de paradigmas de programação foi a falta de linguagens de programação de alto nível, pois até a década de 50 os computadores eram programados usando linguagem de programação de baixo nível, assim problemas decorrentes nestes programas eram difíceis de serem detectados.

Cada paradigma de programação conta com uma estruturação, essa estruturação é escolhida de acordo com a necessidade do desenvolvedor, assim essa estrutura pode ajudar no desenvolvimento de software.

2 PARADIGMA DE PROGRAMAÇÃO PROCEDURAL

O paradigma de programação procedural é baseado em procedimentos de chamadas, que podem ser funções, rotinas, sub-rotinas e métodos.

Este paradigma exige que o programa deve seguir alguns passos para que ele atinja o resultado esperado pelo desenvolvedor.

É possível reutilizar código sem ter a necessidade de ter que copiar e colar, detêm também de modularização e de uma coleção de comandos para a gestão de fluxo ("goto", "jump", etc). Os procedimentos deste paradigma podem chamar a outros procedimentos quanto podem chamar a si mesmos (recursão).

A modularidade do paradigma procedural é usada para a organização de código, pois durante o desenvolvimento de programas de computadores, há softwares mais complexos e que necessitam de mais blocos de códigos, a modularização é um conceito que busca pegar os trechos de códigos e separar em diferentes arquivos afim de unir todos estes arquivos por métodos de inclusão em um arquivo principal, estes procedimentos que foram modularizados em outros arquivos podem ser chamados no arquivo principal através de inclusões e de chamadas de procedimentos ("procedure calls"). A modularização também ajuda nas colisões

de variáveis computacionais, pois impede que variáveis com o mesmo nome possam ser chamadas em outros locais do programa.

Para uma linguagem de programação ter paradigmas de procedimentos, é necessário que ela aceite que suas funções (procedimentos) possam ser chamadas por si mesmas (recursão) e aceite os métodos de modularização.

Algumas linguagens de programação atuais e mais utilizadas que contem traços de programação procedural são: C, C++, Lua, Pascal, Python, Visual Basic, MATLAB, Fortran, COBOL, Basic, Ada, ECMAScript, entre outras.

3 PARADIGMA DE PROGRAMAÇÃO ORIENTADA A OBJETOS

Orientação a objetos é o "processo conceitual independente da linguagem de programação[...]. O desenvolvimento baseado em objetos de uma forma fundamental de pensar e não uma técnica de programação" (RUMBAUGH, et al., 1995, p. 5).

A programação orientada a objetos busca abstrair objetos do "mundo real" para a programação, na questão de semântica o paradigma de programação orientada a objetos é a melhor forma de transformar problemas "reais" para problemas computacionais.

Dentro da orientação a objeto os objetos são uma forma de representar dados da realidade na computação, como por exemplo árvores, veículos, casas e por assim em diante. A modelagem dos objetos é feita dentro de classes, dentro dessas classes há um conjunto de procedimentos e métodos que podem ser invocados por instancias.

As classes da programação orientada podem herdar atributos de outras classes, recebendo assim então esses métodos e atributos que podem ser usados de forma a serem mutáveis.

Para poder usar as classes, é necessário que dentro de nosso código fonte do software fazemos a instanciação desta classe, que fica armazenada normalmente em uma variável computacional, esta variável computacional tem alguns atributos definidos dentro das classes, como acessos privados, protegidos e públicos dos métodos e dados. Assim de acordo com o nível de acesso podemos acessar dados e acessar alguns métodos procedurais públicos ou privados.

Também podemos modelar nossos métodos como polimorfismo, fazendo a herança de classes e fazendo chamadas em outras classes sem alterar os dados daquele método procedural.

Algumas linguagens de programação mais populares que utilizam o conceito de paradigmas de orientação a objeto são: C++, ECMASCRIPT, PHP, Java, Visual Basic, Python e Perl.

4 PARADIGMA DE PROGRAMAÇÃO FUNCIONAL

A programação funcional é um paradigma de programação que busca tratar a aplicação com sua base principal em funções. Normalmente as funções da programação funcional sempre retornam algum valor, mais nem sempre podem apresentar parâmetros.

O paradigma de programação funcional é voltado principalmente para aplicações no campo de estudos da matemática e da física, mas de um tempo para cá o mercado de trabalho vem adotando os paradigmas de programação funcional, pois é uma linguagem que exige menos blocos de códigos para fazer determinadas tarefas e isso reduz facilita bastante na correção de erros e na manutenção de software.

Linguagens de programação funcional tem seu funcionamento rodando em base de interpretadores, que estão aptos a interpretar e construir uma resposta (GLASER, HARTEL et al., 2000).

5 IMPORTANCIA DOS PARADIGMAS DE PROGRAMAÇÃO

Os paradigmas de programação são fatores decisivos na elaboração de um software, pois através deles podemos determinar qual é a melhor solução e a mais viável em nível de estruturação para a construção do software.

Em algumas linguagens de programação, pode haver o caso de ter mais de 1 paradigma de programação, como é o caso do Javascript, que contem paradigmas funcionais, orientados a objetos e procedurais.

6 CONCLUSÃO

A escolha de um paradigma de programação é baseada no ciclo de vida do software, pois é de suma importância que ele seja escalado para poder receber novas funcionalidades, pois com o passar do tempo, alguns métodos utilizados podem ficar obsoletos no mercado de trabalho então é essencial a escolha de um paradigma de programação que se adapte corretamente as mudanças que esse software possa vir a sofrer.

7 REFERÊNCIAS

RUMBAUGH, James et al. *Modelagem e projetos baseados em objetos*. Rio de Janeiro, Campus, 1994.

TUCKER, A. B.; NOONAN, R. M *Programming Languages: Principales and Paradigms*, International Edition, Mc Graw Hill 2001.

SENAGA, Marcelo. *Programação Funcional: Código limpo e padrões de projetos*, Disponível em: <<http://www.devmedia.com.br/programacao-funcional-codigo-limpo-e-padroes-de-projeto/32902>> Acesso em: 30 de jun. de 16.

FRANCISCO, Omero. *Introdução a programação estruturada*, Disponível em: <<http://www.devmedia.com.br/introducao-a-programacao-estruturada/24951>> Acesso em: 30 de jun. de 16.