**LINGUAGEM PROCEDURAL E LINGUAGEM ORIENTADA A OBJETOS**

A linguagem estruturada é um código procedural, o conjunto de procedimentos e funções delimitam o programa. A rotina, o procedimento, é o “cœur” de um programa baseado na estrutura procedural, enquanto os dados manipulados ficam em segundo plano.

“Na programação procedural, as linhas do código são projetadas em torno das funções. Assim sendo, estas são mais importantes neste tipo de programação que na POO, na qual os objetos são mais importantes por serem o núcleo em torno do qual se aglutinam os códigos. Dessa forma, em vez de passar um objeto para a função, como é realizado na linguagem procedural, ele é utilizado para chamá-la de modo que a função fique vinculada ao objeto.” (DIOR FILHO; 1996)

Na programação orientada a objetos, os códigos tendem a ser mais enxutos em casos de funções de múltiplas possibilidades, através do advento das classes, amplamente utilizada em linguagens como JAVA. Desta forma podemos acionar um método, um procedimento, ou até mesmo uma função caso o programador deseje retornar um valor ao ponto da chamada.

Dentro de um programa que simule os sistemas integrados de um carro, poderíamos ter uma classe chamada “Sistema Hidráulico” onde teríamos atributos como “pressão do fluído hidráulico”. Este atributo interage com um atributo dentro da classe “Motor”, pois o atributo “velocidade do motor” irá interagir com o atributo “pressão do fluído”. Se a velocidade do motor aumentar, a pressão do fluído aumentará proporcionalmente. Dentro da mesma classe “Sistema Hidráulico”, poderíamos ter outros atributos como “volante” que aciona uma “válvula de distribuição” (que também pode ser outro atributo) deste fluído. O fluído aumenta a pressão de um dos lados do êmbolo, por sua vez a pressão aplicada no êmbolo depende da força que o motorista aplica no volante.

“Os objetos se parecem com os mini-programas pois podem conter tanto dados como métodos, permitindo que possam ser usados para criar objetos mais complexos. Como afirma WEISKAMP ET AL. (1993) “...muito parecido com o uso dado a transistores e redes na montagem de um circuito integrado...”. Os objetos e as classes ampliam o conceito de tipos de dados abstratos ao somar o conceito de herança. As classes podem organizar-se numa árvore de herança hierárquica. As classes que se encontram nos níveis mais baixos na árvore têm acesso e podem usar dados e comportamento associados com classes mais altas da árvore, portanto, tais classes herdam seu comportamento das classes pais.” (SALDAÑA,2006)

**ENSAIO E CONCLUSÃO**

Me parece que a programação orientada à objetos favorece os conceitos basilares da resolução de problemas. O principal conceito é a **decomposição**, o fato de conseguirmos separar as classes e seus atributos de um problema maior nos ajuda a **abstrair** e também a **reconhecer os padrões** com mais facilidade dentro das classes. Colocar o objeto em evidência nos leva a decompor o problema. Enxergar os atributos de uma classe nos sugerem funções e procedimentos que podem ser utilizados e também como eles podem se relacionar com outros objetos. Eu até arriscaria a dizer que a POO nos aproxima a uma linguagem mais natural; se a pressão de um fluído hidráulico, um atributo, aumenta ou diminui de acordo com o giro do motor, outro atributo de outra classe, então podemos pensar em uma função que leve em conta a proporcionalidade equitativa destas variáveis. Na mesma tese de mestrado, que baseei este pequeno trabalho, o autor ressalta:

“Nos anos 70, os cientistas das linguagens de computadores voltaram aos conceitos básicos. Eles argumentavam, “E se os dados se tornaram foco principal? Pois, vivemos num mundo de objetos e não de procedimentos”. Observando ao redor uma variedade de objetos cada um tem seu próprio conjunto de características, traços e funcionalidades.” (SALDAÑA, 2006, p. 11)

Eu me pergunto se realmente foi apenas esta observação dos cientistas que levaram a uma mudança de paradigma na escrita e estrutura dos códigos, será que eles queriam apenas evidenciar os objetos? Eu acredito que os mesmos, talvez, buscassem, mesmo que inconscientemente, uma tendência de escrita do código a uma linguagem mais natural e se distanciar de uma linguagem de baixo nível, como de um Assembly. Não posso cravar esta hipótese, talvez eu devesse pesquisar mais sobre o assunto.

**Bibliografia:**

Desenvolvimento de um programa orientado por objetos para análise de escoamento em condutos forçados / Edwin Antonio Aranda Saldaña. Campinas, SP, 2006, Capítulo 2 – Revisão Bibliográfica.