UENF – Ciência da Computação

João Víttor Vieira Pinto

Resumo do Artigo: “What is Object Oriented Programming”

**1-Introduction**

Nem todas linguagens de programação podem ser orientadas a objeto. Alguns exemplos de linguagens que são orientadas a objeto: APL, Ada, Clu, C++, LOOPS e Smalltalk. Nesse resumo o principal foco é na distinção da programação orientada a objeto da abstração de dados e dos mecanismos que são essenciais para dar um suporte a esses vários estilos de programação.

**2-Programming Paradigms**

Programação orientada em objetos é, basicamente, uma técnica (razoavelmente mais fácil, mais segura e eficiente) para programar, deixando mais conveniente a escrita de alguns estilos de programação. Além disso, esse suporte também vem de forma mais sutil, como a verificação de compile-time e/ou run-time checks contra desvios não intencionais do paradigma. Uma linguagem não é necessariamente melhor do que a outra só por ter um recurso que a outra não tem.

**2.1-Procedural Programming**

O foco é no design do processamento, o algoritmo precisa fazer a computação desejada. O Fortran é a linguagem processual original, sendo que C e Pascal foram criadas com a mesma ideia. As criações de funções são importantes para a organização do algoritmo.

**2.2-Data Hiding**

Com o passar dos anos, o tamanho dos programas foram aumentando, tornando-se necessário a manipulação dos dados, chamando assim de módulo. As técnicas para o design de bons procedimentos são aplicadas para cada um processo do módulo. Os problemas principais que devem possuir uma boa solução são: Primeiro, providenciar uma interface para o stack; se assegurar que a apresentação da stack só pode ser acessada pela interface do usuário; se assegurar que a stack é inicializada antes do primeiro uso.

2.3-Data Abstraction

A programação com módulos leva à centralização de todos os dados de um tipo sob o controle de um “gerenciador de modulo”. Isso é uma grande melhoria em relação à bagunça tradicional, mas os tipos implementados são claramente muito diferentes dos tipos integrados em uma linguagem. Cada gerenciador de módulo de tipo deve definir um mecanismo separado para a criação de "variáveis" de seu tipo.

2.4-Problems with Data Abstraction

Um tipo “dado abstrato” é como se fosse uma caixa preta. Uma vez criado, ele não interage com o resto do programa. Não há maneira de adaptá-lo a novos usos, exceto modificando sua definição. Isso pode levar a uma severa e chata inflexibilidade.

2.5-Object-Oriented Programming

O problema principal é que não há distinção entre as propriedades gerais de uma qualquer forma e as propriedades de uma forma específica. Expressar essa distinção e tirar proveito dela define programação orientada a objetos. Uma linguagem com construções que permite que essa distinção seja expressada e usada oferecendo suporte. Outras linguagens não. A quantidade de semelhança entre tipos que podem ser explorados usando herança e funções virtuais é o teste decisivo da aplicabilidade de programação orientada a objetos para uma área de aplicação. Encontrar pontos comuns entre os tipos em um sistema não é um processo trivial.