***UM BREVE HISTÓRICO DE LINGUAGENS DE PROGRAMAÇÃO***

Os caminhos da programação têm mudado dramaticamente desde a invenção do

computador. A primeira razão para as mudanças é acomodar o aumento da complexidade dosprogramas. Por exemplo, quando os computadores foram inventados, a programação era feita por chaveamentos em instruções binárias de máquina, usando-se o painel frontal. Enquanto os programas continham somente algumas centenas de instruções, esse método funcionava. Quando cresceram, a linguagem assembly foi inventada para permitir a um programador manipular complexidades. A primeira linguagem de alto nível difundida foi, obviamente, FORTRAN. Ainda que o FORTRAN tenha dado um primeiro passo bastante considerável, é uma linguagem que somente torna os programas mais claros e fáceis de entender do que o assembly, sem introduzir mudanças consideráveis no estilo de programação.Nos anos 60 nasceu a programação estruturada. Esse é o método estimulado por linguagens como C e Pascal. Usando-se linguagens estruturadas, foi possível, pela primeira vez, escrever programas moderadamente complexos de maneira razoavelmente fácil. Entretanto, com programação estruturada, quando um projeto atinge um certo tamanho, torna-se extremamente difícil e muito custoso efetuar sua manutenção e fazer qualquer modificação. A cada marco no desenvolvimento da programação, métodos foram criados para permitir ao programador tratar com complexidades incrivelmente grandes. Cada passo combinava os melhores elementos dos métodos anteriores com elementos mais avançados. Atualmente, muitos projetos estão próximos ou no ponto em que o tratamento estruturado não mais funciona. Para resolver esse problema, a programação orientada a objetos foi criada.

A programação orientada a objetos aproveitou as melhores idéias da programação

estruturada e combinou-as com novos conceitos poderosos que levam a ver na tarefa de

programação uma nova luz, permitindo que um problema seja mais facilmente decomposto em subgrupos relacionados. Então, usando-se a linguagem, pode-se traduzir esses subgrupos em objetos.

A primeira linguagem a incorporar facilidades para definir classes de objetos genéricos na forma de uma hierarquia de classes e subclasses foi a linguagem Simula, que foi idealizada em 1966, na Noruega, como uma extensão da linguagem ALGOL 60.

Uma classe em Simula, é um módulo englobando a definição da estrutura e do

comportamento comuns a todas as suas instâncias (objetos). Como o nome indica, é uma

linguagem adequada à programação de simulações de sistemas que podem ser modelados pela interação de um grande número de objetos distintos.

As idéias de Simula serviram de base para as propostas de utilização de Tipos Abstratos

de Dados, e também para Smalltalk. Smalltalk foi desenvolvida no Centro de Pesquisas da Xerox durante a década de 70, e incorporou, além das idéias de Simula, um outro conceito importante, devido a Alan Kay, um de seus idealizadores: o princípio de objetos ativos, prontos a “reagir” a “mensagens” que ativam “comportamentos” específicos do objeto. Ou seja, os objetos em Smalltalk deixam de ser meros “dados” manipulados por “programas”, e passam a ser encarados como “processadores idealizados” individuais e independentes, aos quais podem ser transmitidos comandos em forma de “mensagens”.

Outras linguagens orientadas para objetos tem sido desenvolvidas. Notadamente C++,

uma extensão de C, Objetive-C, outra extensão de C, menos popular que a anterior, Pascal

orientado a objetos, Eiffel e mais recentemente, no Brasil, TOOL.

Além da Xerox, que criou a ParcPlace Systems especialmente para comercializar

Smalltalk-80 e seus sucedâneos (objectWorks), a Digitalk lançou em 1986 uma versão de

Smalltalk para ambiente DOS, e mais recentemente a versão para Windows, o que contribuiu para uma maior difusão da linguagem [Digitalk].Smalltalk, uma das mais populares linguagens orientadas a objetos, assim como outras linguagens orientadas para objetos, tem sido usada em aplicações variadas onde a ênfase está na simulação de modelos de sistemas, como automação de escritórios, animação gráfica, informática educativa, instrumentos virtuais, editores de texto e bancos de dados em geral,

entre outras. Tais aplicações diferem substancialmente daquelas em que a ênfase está na

resolução de problemas através de algoritmos, tais como problemas de busca, otimização e resolução numérica de equações. Para essas aplicações, é mais adequado o uso de linguagens algorítmicas convencionais, como Pascal, Algol e Fortran.