ESI

Matheus Andreoli Vieira 2ºY Nº20

**Programação Orientada a Objeto**

**COMERN**

**C:** Confiável. Se alguma parte for alterada, a outra não será alterada por causa do isolamento de partes, gerando um Software seguro. Exemplo: se eu quiser alterar algo no meu celular, na agenda, por exemplo, nada mais será alterado.

**O:** Oportuno. As partes podem ser desenvolvidas paralelamente, não precisam ser feitas no mesmo momento. Exemplo: um programa que faz vários cálculos, cada um pode ser feito em determinado momento.

**M:** Manutenível. Se alguma alteração for feita, uma atualização, as partes que usam aquele objeto serão beneficiadas. Exemplo: um jogo vai ter uma alteração na mira, apenas a mira e o que está ligado a mira serão alterados.

**E:** Extensível. É possível acrescentar ao objeto utilizando o que já está nele. Exemplo: uma impressora que não tira cópias, é possível acrescentar essa opção nela utilizando o que já havia nela, sem apagar nada.

**R:** Reutilizável. Uma classe criada pode ser usada várias vezes. Exemplo: uma calculadora, pode fazer um cálculo diversas vezes.

**N:** Natural. Deve ser algo de fácil entendimento.

**Projetos de Alan Kay**

Smalltalk: uma linguagem Orientada a Objeto;

Fundou em 2001 o *Viewpoints Research Institute*, uma organização sem fins lucrativos dedicada às crianças, ao aprendizado e ao desenvolvimento de software avançado.

Dispositivos baseados na ideia de Alan Kay

Notebooks, laptops, tablets.

**Evolução das Linguagens de Programação**

**1-Baixo Nível:** Os comandos tinham que ser dados de forma que o computador entendesse.

**2-Programação Linear:** Os comandos se tornaram compreensíveis pelos programadores. Os comandos eram executados conforme fossem escritos, linha por linha, de cima para baixo.

**3-Programação Estruturada:** permitia pegar trechos feitos em linear e utilizá-los fora de ordem.

**4-Programação Modular:** Permitia criar pequenos módulos estruturados e coloca-los em cápsulas que poderiam compor sistemas.

**5-Programação Orientada a Objeto:** Facilita o reuso e manutenção do seu código.

**Diferenças entre linguagens OO e não OO.**

Em uma linguagem OO é muito mais fácil a reutilização de um código, você pode usá-lo várias vezes, já na outra linguagem isso não é possível, deve-se escrever o código de novo; Na não OO o foco é em como a tarefa será feita e não no que deve ser feito, já na OO ela visa fazer o melhor caminho para execução do código; Segundo pesquisas que fiz, a linguagem OO é de difícil entendimento, em relação a não OO.