# Programação Procedural

**Definição**: Programação estruturada é uma forma de programação de computadores que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: sequência, decisão e iteração. Tendo, na prática, sido transformada na Programação modular, a Programação estruturada orienta os programadores para a criação de estruturas simples em seus programas, usando as sub-rotinas e as funções. Foi a forma dominante na criação de software entre a programação linear e a programação orientada por objetos. Apesar de ter sido sucedida pela programação orientada por objetos, pode-se dizer que a programação estruturada ainda é marcantemente influente, uma vez que grande parte das pessoas ainda aprendem programação através dela. Porém, a orientação a objetos superou o uso das linguagens estruturadas no mercado.

Características:

* Decomposição gradativa dos programas ao nível fundamental (método dos refinamentos sucessivos, desenvolvimento top-down).
* Programação orientada a procedimentos: subprogramas = blocos estruturados de códigos (procedimentos, funções ou módulos); a comunicação entre os blocos se faz utilizando variáveis globais e pela passagem de dados através de parâmetros; os dados são processados nos blocos e migram de um bloco para outro, como mostra a Figura 1, através de variáveis globais, parâmetros passados por referência e expressão retornada pela função (através do comando return na linguagem C); a execução de um programa é caracterizada pelo acionamento de um bloco de código. Obs.: a utilização de variáveis globais não constitui uma boa prática de programação (escopo muito grande).
* Tipo abstrato de dados = modelo matemático + operações.

**Prós:** Provê um melhor controle sobre o fluxo de execução do código, quando comparada  
com a programação imperativa.  
É fácil de se entender, sendo amplamente usada em cursos introdutórios de programação.

**Contras**: Ainda se foca em como a tarefa deve ser feita e não em o que deve ser feito.  
Tende a gerar códigos confusos, onde tratamento dos dados são misturados com  
o comportamento do programa.

# Programação Orientada a Objetos

**Definição:** Orientação a objetos, também conhecida como Programação Orientada a Objetos (POO) é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos. Em alguns contextos, prefere-se usar modelagem orientada ao objeto, em vez de programação. Na qualidade de método de modelagem, é tida como a melhor estratégia, e mais natural, para se eliminar o "gap semântico", dificuldade recorrente no processo de modelar o mundo real, no domínio do problema, em um conjunto de componentes de software que seja o mais fiel na sua representação deste domínio. A análise e projeto orientados a objetos tem como meta identificar o melhor conjunto de objetos para descrever um sistema de software.

Características:

* Classe: É o molde para criar objetos. Possui todas as especificações de um grupo deles.  
  Métodos: Definem o comportamento dos objetos, tendo seus nomes normalmente  
  definidos por verbos.
* Abstração: É a habilidade de se concentrar nos principais aspectos de um grupo de  
  objetos, em vez de se preocupar com as suas especificações.  
  Encapsulamento: É a habilidade de esconder de outros objetos, as características  
  intrínsecas de um dado objeto. Toda a comunicação inter objetos deve ser realizada  
  via métodos. Um objeto não deve ser capaz de acessar, e tampouco alterar, atributos  
  de outro objeto diretamente.
* Associação: É a habilidade pela qual um objeto utiliza recursos de outro.
* Herança: É a capacidade de criar subclasses a partir de uma superclasse.  
  Essas subclasses herdam, então, todas as características da superclasse.  
  É normalmente definida como uma associação do tipo “é um”, e.g., Cliente é uma Pessoa.
* Polimorfismo: É o princípio pelo qual uma subclasse sobrescreve um comportamento  
  (método) herdado de sua superclasse. Por exemplo, a classe Pessoa implementa o  
  método Comprar, mas a classe Cliente, derivada da classe Pessoa, precisa de algo  
  mais específico, envolvendo mais dados. Então a classe Cliente sobrescreve o  
  método Comprar, tornando-o mais específico.

**Prós:** Provê uma melhor organização do código. Contribui para o reaproveitamento de código.

**Contras:** Não possui o mesmo desempenho de códigos estruturados similares. Seus conceitos são de difícil compreensão se comparados aos conceitos da Programação estruturada.