**Pilares da Orientação a Objeto:**

**Encapsulamento:** Encapsular objetos. Você se preocupa apenas com o que está visível, apenas com o que está público. Algumas coisas irão ficar privadas, e outras públicas, cabe a você saber distinguir o que vai ser público e o que vai ser privado. Isto está fortemente ligado aos Modificadores de Acesso (public, protected, package, private). O objeto ele serve como uma cápsula que “encapsula” os atributos e métodos, gerando um encapsulamento. E com isto, algumas coisas conforme dito serão privadas e outras públicas.

**Herança:** Herança se trata de um reuso de código. Para descobrir se vamos usar Composição ou herança é só usar as frases: Tem um(a) ou é um(a). **Exemplo:** Carro tem um motor. Civic é um carro. No caso, é dividido em classes, onde a classe filho irá herdar os métodos e comportamentos da classe pai. A classe mais genérica, deve ter métodos genéricos que sirvam em todos as classes filhas.

**Polimorfismo:** Dois tipos de polimorfismo: Estático (Sobrecarga) e dinâmico. Polimorfismo estático é quando você possui diversos métodos em uma classe com a mesma assinatura (nome) mas com parâmetros diferentes. Polimorfismo dinâmico, para ter ele precisa ter herança. Ele é definido quando vocês instância um Objeto, onde o tipo dele é um tipo genérico, mas ele é criado com algo mais objetivo. Por exemplo:

Carro c = new Civic();

c = new Ferrari();

Eu consigo mudar o objeto c para uma Ferrari, pois ele é do objeto genérico Carro.

**Abstração:** Pegar um conceito do mundo real, e trazer para o código. Simplificar algo do mundo real e colocar no software. Isto é feito através das classes ,atributos e métodos.