

Java Básico

**FUNDAMENTOS**

Java e sua história

Plataformas

Configurando o ambiente

Escolha sua IDE

**SINTAXE**

Anatomia das classes

Tipos e Variáveis

Operadores

Métodos

Escopo

Palavras reservadas

Documentação

Terminal e Argumentos

**CONTROLE DE FLUXO**

Conceito

# Encapsulamento

Nem tudo precisa ser estar disponível para todos!

Já imaginou, você instalar o MSN Messenger e ao querer enviar uma mensagem, fosse solicitado a você, verificar se o computador está conectado a internet e depois, pedir para você salvar a mensagem no histórico? ou, se ao tentar enviar um SMS pelo celular, primeiro você precisasse consultar manualmente o seu saldo ?

Acredito que não seria uma experiência tão agradável de ser executada, recorrentemente, por nós usuários.

Mesmo ainda sendo necessária algumas etapas, nos processos citados, não é um requisito uma visibilidade pública, isso quer dizer, além da própria classe que possui a responsabilidade de uma determinada ação.

Quanto ao MSN Messenger, para nós, só é relevante saber que podemos e como devemos enviar e receber a mensagem, logo, as demais funcionalidades poderão ser consideradas privadas (private). E é aí que se caracteriza a necessidade do pilar de Encapsulamento. O que esconder ?

Vejamos a refatoração abaixo, da nossa classe MSN Messenger:

UML

Antes

MSNMessenger.java

Depois

## MSNMessenger

- + enviarMensagem() : void
- + receberMensagem() : void
- validarConectadoInternet() : void
- salvarHistoricoMensagem() : void



Previous  
Pilares do POO

Next  
Herança



Last updated 1 year ago