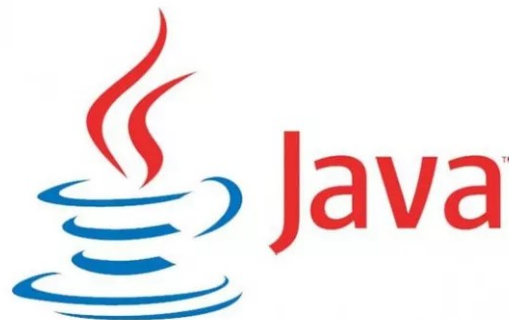


# Orientação a Objetos

Prof. Daniel Di Domenico

[daniel.domenico@ifpr.edu.br](mailto:daniel.domenico@ifpr.edu.br)

Implementando classes com Java



# Objetivos da aula

- **Revisar conceitos de Classes e Objetos em Java:**
  - Modificadores de acesso
  - Encapsulamento (GETs e SETs)
  - Herança e polimorfismo

# Requisitos de software

- **IDE Eclipse**

- Já vem com o Java embutido
- Java deve ser versão 8 ou superior

# Modificadores de acesso

- Padrões de visibilidade utilizados pelos atributos e métodos de uma classe:
  - **public (+)**: qualquer outra parte da aplicação pode visualizar o atributo/método
  - **private (-)**: o atributo/método é visível apenas dentro da classe onde foi declarado
  - **protected (#)**: o atributo/método é visível pela classe onde foi declarado, bem como por suas classes derivadas (classes filhas)
- *Obs.: em Java, os modificadores também podem contemplar os pacotes, porém tais funcionalidades não serão abordados neste disciplina*

# Encapsulamento

- Princípio de controlar o acesso às funcionalidades de uma classe (atributos e métodos):
  - É utilizado para proteger os dados que precisam ser manipulados apenas de forma interna na classe
  - Por princípio, os atributos de uma classe devem ser sempre privados
    - O acesso a eles deve ser concedido por métodos GETs e SETs

# Herança

- Permite que as características de uma classe (superclasse) sejam compartilhadas com suas classes derivadas (filhas):
  - A classe filha vai herdar todos os atributos e métodos da classe pai
  - Os métodos da classe pai podem ser sobrescritos nas classes filhas:
    - Tal propriedade é chamada de **polimorfismo**
    - Sempre o método da classe mais específica é invocado
  - Em Java, todas as classes criadas são filhos da classe Object:
    - Método toString() padrão é implementado nesta classe, mas pode ser sobrescrito utilizando polimorfismo

# Herança

- Em Java, utiliza-se a palavra ***extends*** para definir a herança:

```
public class ClientePF extends Cliente {  
  
}
```