

# Orientação a Objetos

Prof. Daniel Di Domenico

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

Persistência de dados



# Objetivos da aula

- **Persistência de objetos em base de dados**
  - Salvar os objetos de forma persistente
  - Manter os objetos após finalizar a execução do programa
- **Depuração de código**
  - Executar o programa linha a linha

# Requisitos de software

- IDE Eclipse
  - Já vem com o Java embutido
  - Java deve ser versão 8 ou superior
- Banco de dados MySQL
- Aplicação cliente para acesso ao MySQL
  - phpMyAdmin ou outro
- Drive do MySQL para Java (.jar)
  - <https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.30>

# Conexão ao banco de dados

- Em Java, utiliza-se a API **JDBC**
  - *Java Database Connectivity*
  - Suporta conexão em Java a diversos bancos, entre eles o MySQL
  - Requer o drive do banco de dados (.jar)
  - Pacote de bibliotecas: java.sql.\*

```
dbConn =  
    (Connection) DriverManager.getConnection(  
        "jdbc:mysql://<host>:<porta>/base",  
        "usuario", "senha");
```

# Persistência: inserir

- Exemplo de código para **inserir** um registro na base de dados

```
String sql = "INSERT INTO tabela (campo1, campo2)  
VALUES (?, ?)";
```

```
PreparedStatement ps =  
    conn.prepareStatement(sql);
```

```
ps.setString(1, "valor"); //Seta os parâmetros  
ps.setInteger(2, 1254);
```

```
ps.executeUpdate(); //Executa a statement
```

# Persistência: buscar

- Exemplo de código para **buscar** registros da base de dados

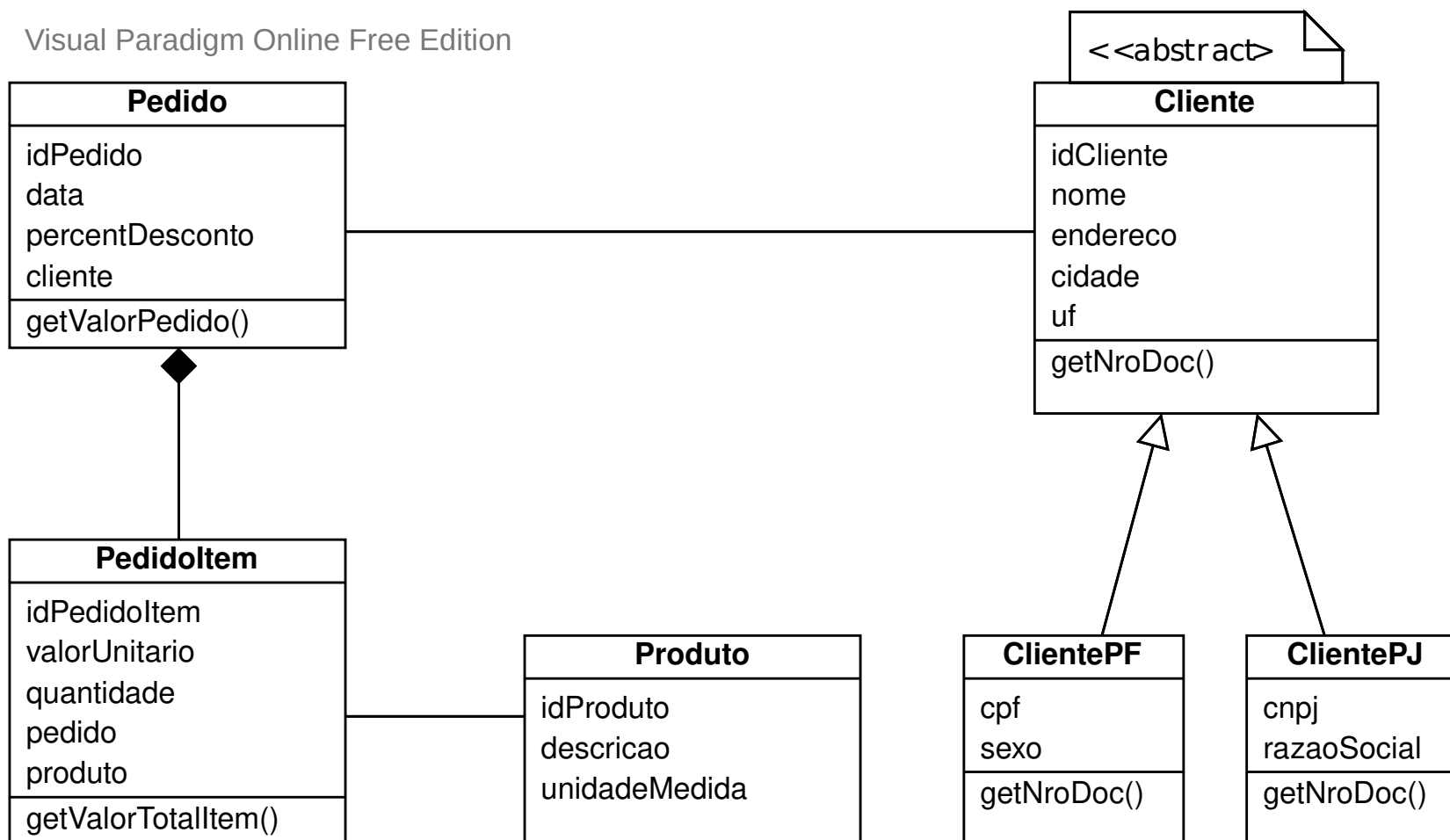
```
String sql = "SELECT campo1, campo2 FROM tab";

PreparedStatement ps =
    conn.prepareStatement(sql);
//Executa a statement, capturando o retorno
ResultSet rs = ps.executeQuery();

while(rs.next()) { //Percorre os resultados
    String c1 = rs.getString("campo1");
    String c2 = rs.getInt("campo2");
    System.out.println(c1 + " " + c2);
}
```

# Diagrama de classes proposto

Visual Paradigm Online Free Edition



# Depuração de código (Debug)

- Auxilia o desenvolvedor a identificar problemas no código-fonte de uma aplicação
- A IDE Eclipse permite realizar depuração de programas Java:
  - Passo 1: adicionar o(s) ponto(s) de parada
  - Passo 2: executar o código linha a linha por meio dos comandos (teclas de atalho):
    - **F5**: Fluxo da execução vai para a próxima instrução, entrando no método que está sendo chamado
    - **F6**: Fluxo da execução vai para a próxima instrução mantendo-se no método que está sendo executado
    - **F8**: Fluxo da execução é resumido (encerra a depuração). No entanto, o fluxo pode ser direcionado para outro ponto de parada futuro