

“Diretrizes, Padrões e código para aplicativos Java de ponta a ponta”

Uma primeira visita ao Design Pattern - DAO

Dieval Guizelini

Baseado em:

<http://www.oracle.com/technetwork/java/patterns-139816.html>

Sobre o BluePrints

(planta, diagrama, desenho técnico...)



Menu



Sign In ▾



Country/Region ▾



Call

Oracle Technology Network / Java

Java SE
Java EE
Java ME
Java SE Subscription
Java Embedded
Java Card
Java TV
Community
Java Magazine

Java BluePrints

Guidelines, Patterns, and code for end-to-end Java applications.

Are you developing enterprise applications? Web services? Wireless services? Are robustness, scalability, and portability important to you? Java technology can help, and we have the blueprints for you. Use the links below to find out more.

BluePrints Online

- [The Java Pet Store Reference Application Released](#)
The Java BluePrints program released an [early access Java Pet Store Reference Application](#). This application is designed to illustrate how the Java EE 5 Platform can be used to design and develop an AJAX-enabled Web application. The application comes with full source-code available under a BSD-style license, so you can experiment with it and use it in your own applications.
- [Java BluePrints Solutions Catalog for Java EE 5 Early Access now available](#)
Check it out! The early access release of the Java BluePrints Solutions catalog for Java EE 5 is now available. [Download it now](#) and run it on the [GlassFish](#) Java EE 5 application server.
- [Java Adventure Builder Reference application](#)
Check out the latest enhancements to this sample application which illustrates BluePrints guidelines and patterns for Web services.
- [Java BluePrints AJAX Page](#)
Check this out. Java BluePrints has organized all of its AJAX for Java Platform, Enterprise Edition (Java EE) content in one location.
- [Java BluePrints Solutions Catalog for J2EE](#)
The Java BluePrints Solutions Catalog presents the latest BluePrints guidelines on designing AJAX (Asynchronous JavaScript and XML) applications on the J2EE platform, service-oriented architectures (SOA), JavaServer Faces technology, business tier issues, and other J2EE platform topics. Each entry follows a problem-solution format which covers strategies, patterns, and design recommendations for a given problem.

Java SDKs and Tools

- ▾ [Java SE](#)
- ▾ [Java EE and Glassfish](#)
- ▾ [Java ME](#)
- ▾ [Java Card](#)
- ▾ [NetBeans IDE](#)
- ▾ [Java Mission Control](#)

Java Resources

- ▾ [Java APIs](#)
- ▾ [Technical Articles](#)
- ▾ [Demos and Videos](#)
- ▾ [Forums](#)
- ▾ [Java Magazine](#)
- ▾ [Developer Training](#)
- ▾ [Tutorials](#)
- ▾ [Java.com](#)



- **The Java BluePrints book Designing Web Services with J2EE JAX-RPC, SOAP, and XML Technologies**
is now available for free download in PDF format. This book provides the guidelines, patterns, and real-world examples architects and developers need for designing and integrating enterprise-level Web services using the Java 2 Platform, Enterprise Edition (J2EE). The book provides insight into the details behind the Java Adventure Builder Reference application.
- **Designing Web Services with J2EE: JAX-RPC, SOAP, and XML Technologies**
Online version of the latest book on building Service Oriented Architectures using J2EE technologies from the J2EE BluePrints Team.
- **BluePrints in J2EE SDK Samples bundle**
The Java Adventure Builder Reference application and the Java Pet Store Demo are available for download from the J2EE SDK Samples bundle.
- **J2EE Design Patterns**
A palette of patterns you can use when building J2EE applications.
- **Naming Conventions for Enterprise Applications, *Early Access***
This document describes naming conventions for Enterprise Applications.
- **Project Conventions for Enterprise Applications, *Early Access***
This document describes project conventions for Enterprise Applications. The conventions in this document are intended to assist developers with organizing the files and directories associated with an application project in a logical fashion.
- **Designing Enterprise Applications with the J2EE Platform, Second Edition**
Guidelines from the second edition of our best-selling book.
Second Edition: [Online](#) | [Download](#) | [Purchase](#)

Welcome to Core J2EE Patterns!

On this site, you will find the entire Java 2 Platform, Enterprise Edition (J2EE) Pattern catalog from the book *Core J2EE Patterns: Best Practices and Design Strategies* authored by architects from the [Sun Java Center](#). All patterns are published in their entire form from the first edition of the book.

Places to go from here:

- [Core J2EE Patterns Catalog](#) (with all the 15 patterns from the book)
- [Read about the book](#)
- Read the forewords from the book:
 - [Foreword by Grady Booch](#)
 - [Foreword by Martin Fowler](#)
- Read [what experts say](#)
- Download [the book's sample code](#)
- Check out the [J2EE Tutorial](#)
- Check out the [Java BluePrints for the Enterprise](#)
- Learn how the [Java Pet Store](#) sample application implements the Core J2EE Patterns
- Learn about the [Sun Java Center](#)
- Check out other [resources](#)
- [Buy the book](#)
- [Contact Us](#): We love to hear from you!

Last Updated:22-Jul-2002

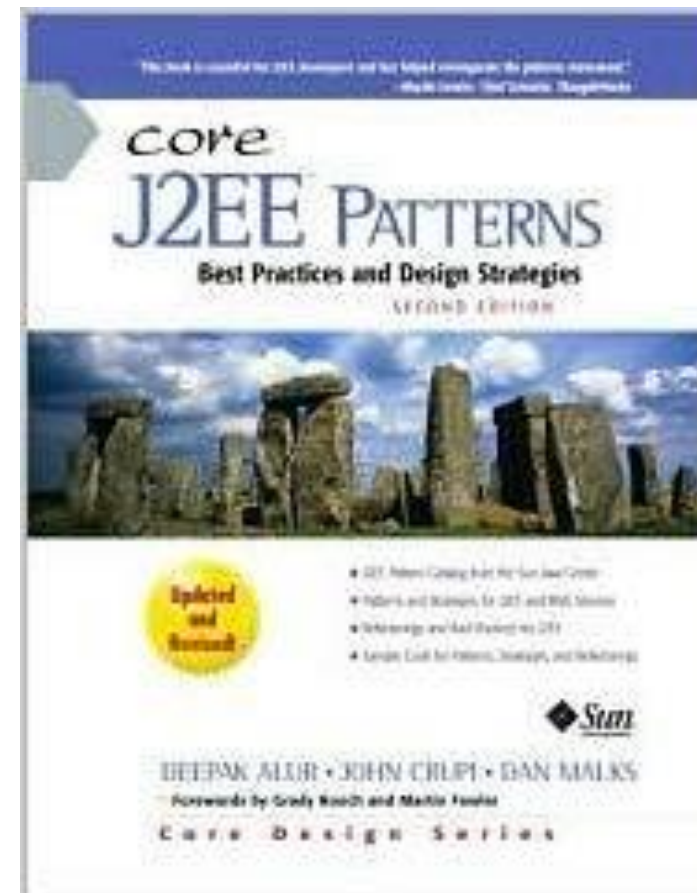
[link](#)

What Experts Say about the book:

"The authors are the first group to extend my work on refactoring into the world of J2EE design. Not just am I grateful...I'm glad they've used their experience to outline how to do these transformations."

Martin Fowler

Chief Scientist, *ThoughtWorks*
(excerpted from [Foreword](#))

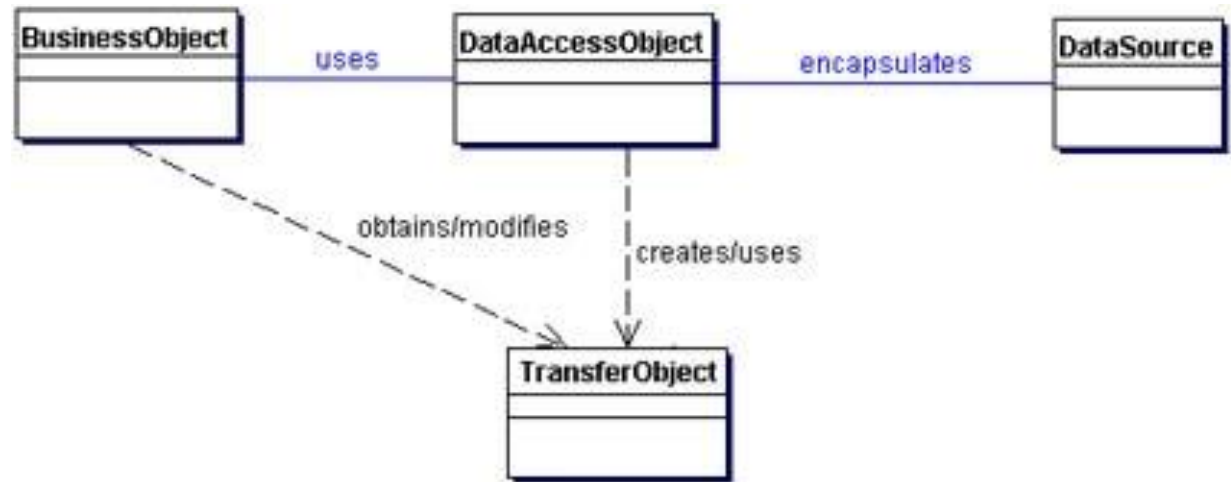


Core J2EE Patterns - Data Access Object

- **Contexto:** O acesso aos dados varia dependendo da origem dos dados. O acesso ao armazenamento persistente, como em um banco de dados, varia muito, dependendo do tipo de armazenamento (bancos de dados relacionais, bancos de dados orientados a objetos, arquivos simples e assim por diante) e da implementação do fornecedor.
- **Problema:** A maioria das aplicações dependem de um mecanismo de persistência de dados (geralmente RDBMS).
- **Solução:** Use um Data Access Object (DAO) para abstrair e encapsular todo o acesso à fonte de dados. O DAO gerencia a conexão com a fonte de dados para obter e armazenar dados.

Data Access Object (DAO)

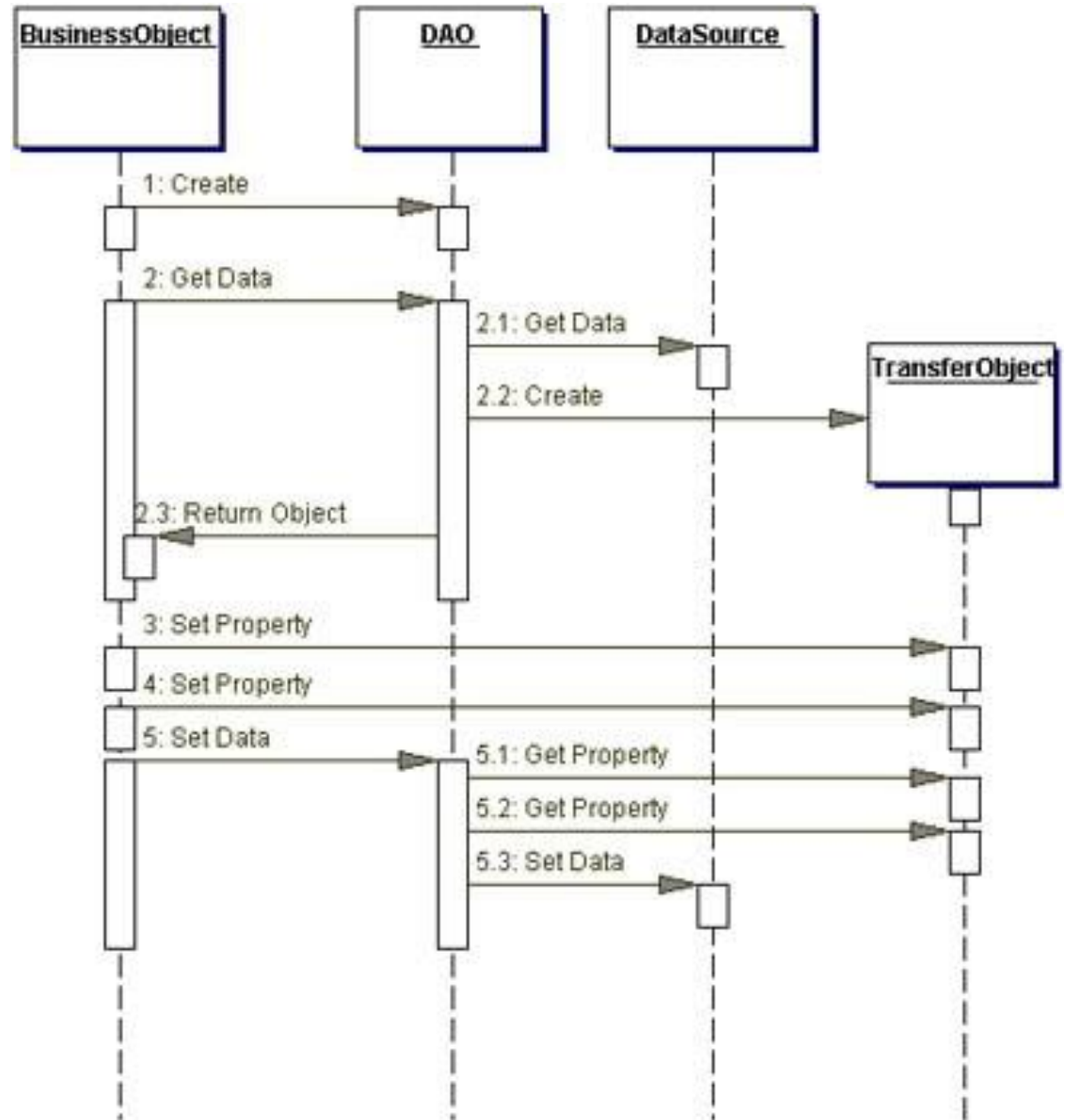
- Estrutura:



Data Access Object (DAO)

Participantes e Responsabilidades:

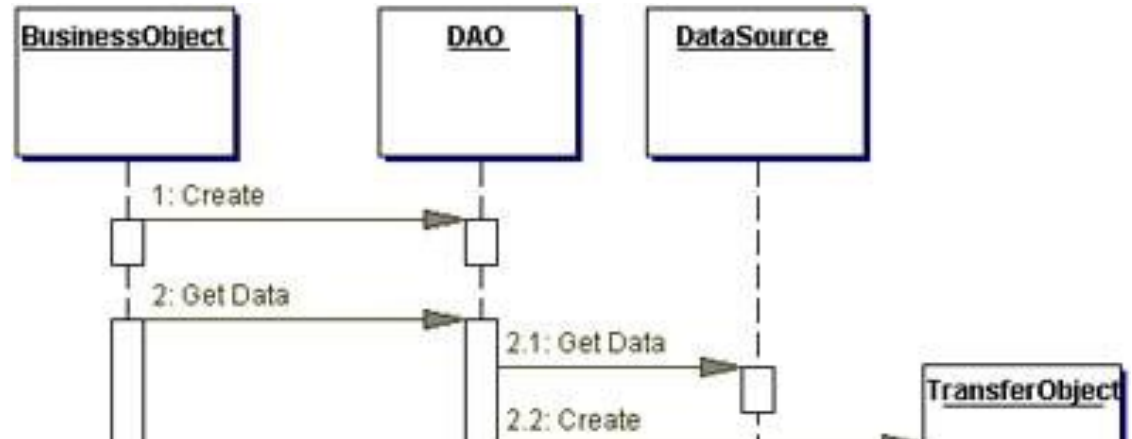
- BusinessObject
- DataAccessObject
- DataSource
- TransferObject



Data Access Object (DAO)

Participantes e Responsabilidades:

- BusinessObject



O BusinessObject representa o cliente de dados.

É o objeto que requer acesso à fonte de dados para obter e armazenar dados.

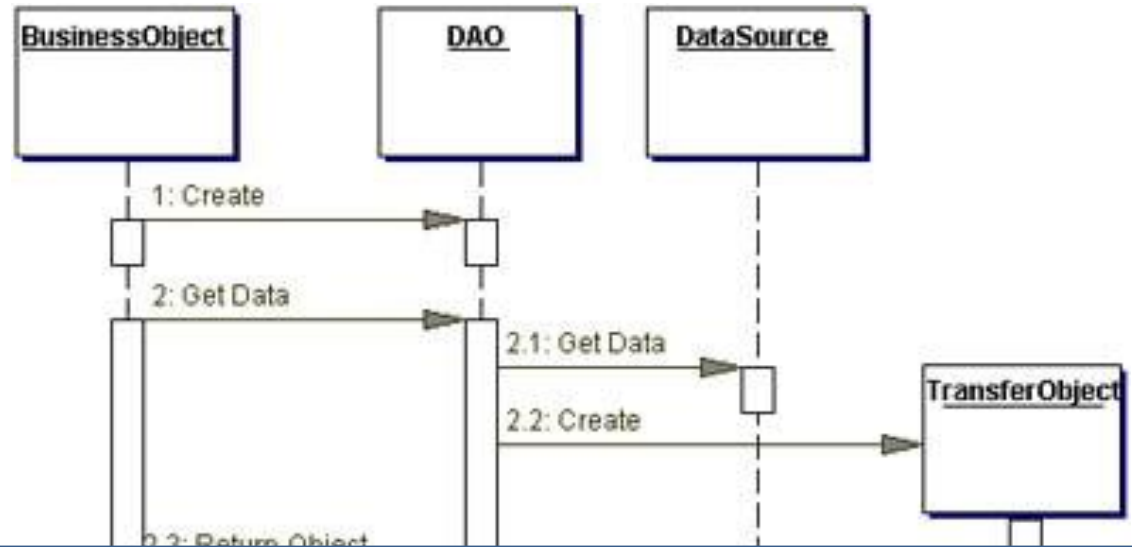
Um BusinessObject pode ser implementado como um bean de sessão, um bean de entidade ou algum outro objeto Java, além de um servlet ou bean auxiliar que acessa a origem de dados.

- DataAccessObject
- DataSource
- TransferObject

Data Access Object (DAO)

Participantes e
Responsabilidades:

- BusinessObject
- DataAccessObject



O DataAccessObject é o objeto principal desse padrão.

O DataAccessObject abstrai a implementação de acesso a dados subjacente para o BusinessObject para permitir o acesso transparente à fonte de dados.

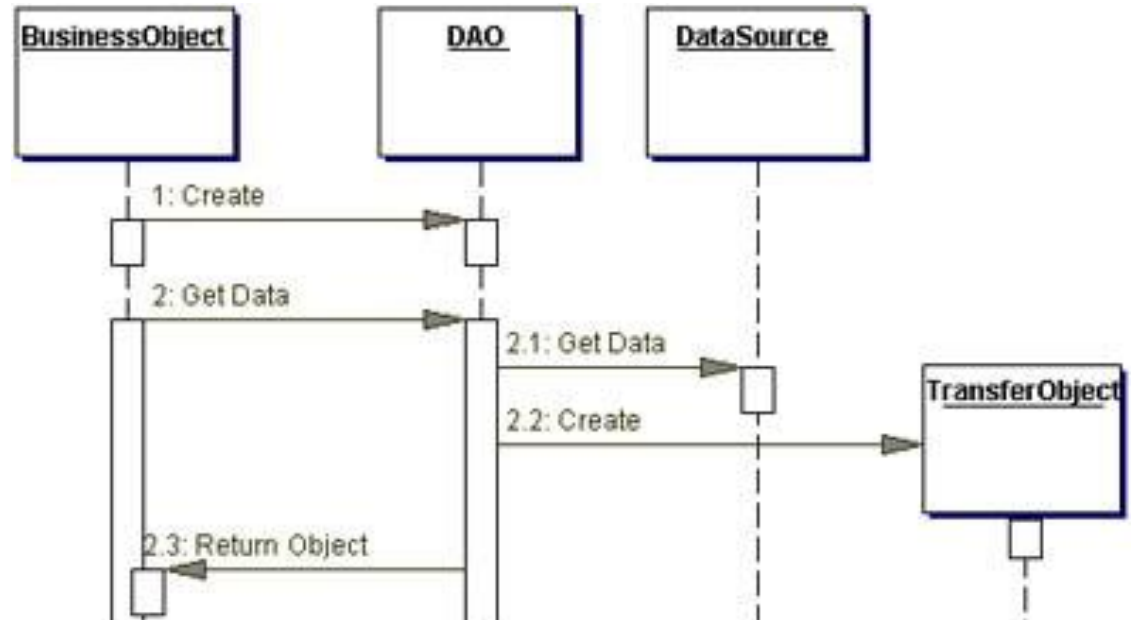
O BusinessObject também delega operações de carregamento e armazenamento de dados para o DataAccessObject.

- DataSource
- TransferObject

Data Access Object (DAO)

Participantes e Responsabilidades:

- BusinessObject
- DataAccessObject
- DataSource



Isso representa uma implementação de fonte de dados.

Uma origem de dados pode ser um banco de dados, como um repositório RDBMS, OODBMS, XML, sistema de arquivos simples e assim por diante.

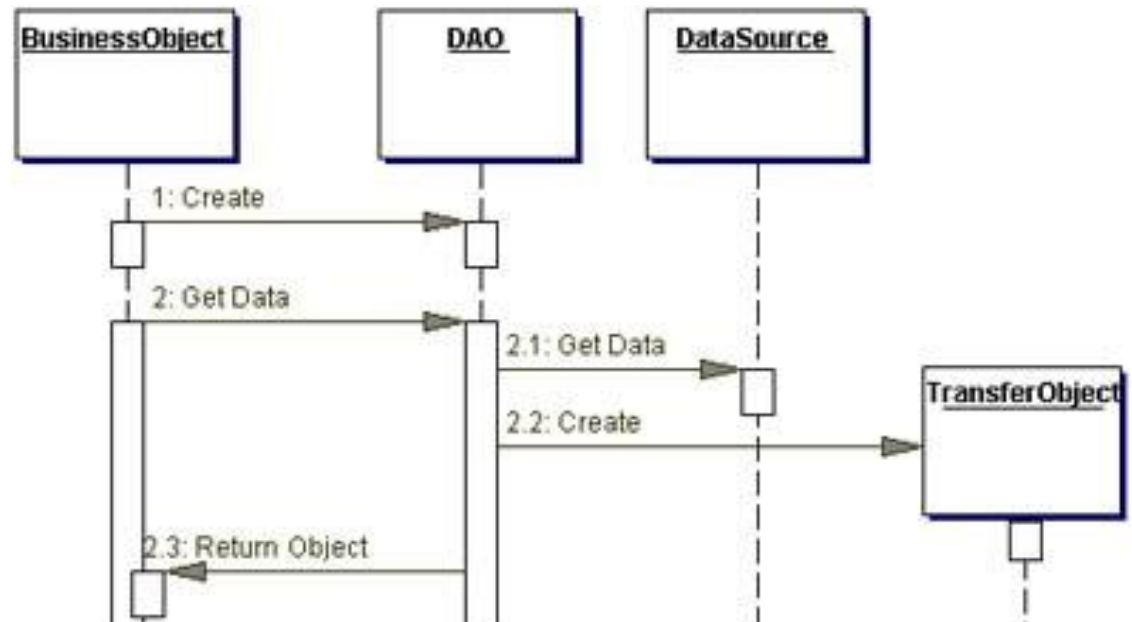
Uma fonte de dados também pode ser outro sistema (legado / mainframe), serviço (serviço B2B ou agência de cartão de crédito) ou algum tipo de repositório (LDAP).

- TransferObject

Data Access Object (DAO)

Participantes e Responsabilidades:

- BusinessObject
- DataAccessObject
- DataSource
- TransferObject

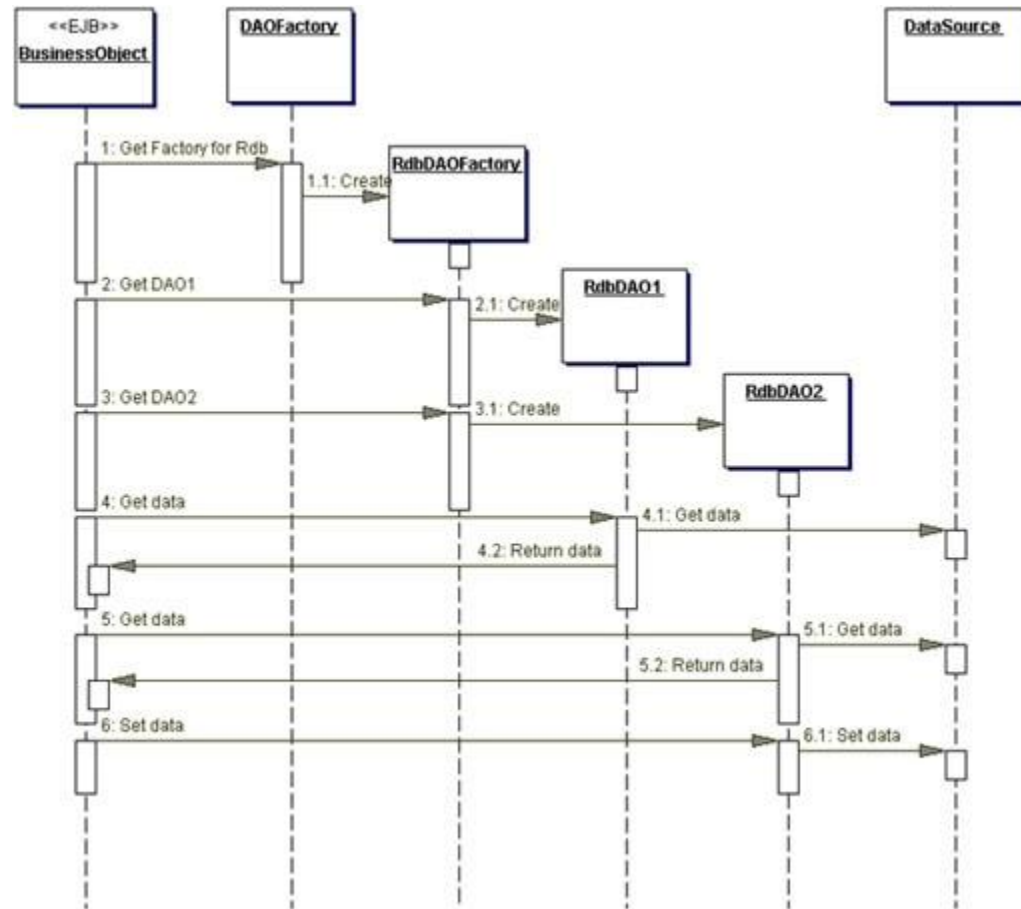
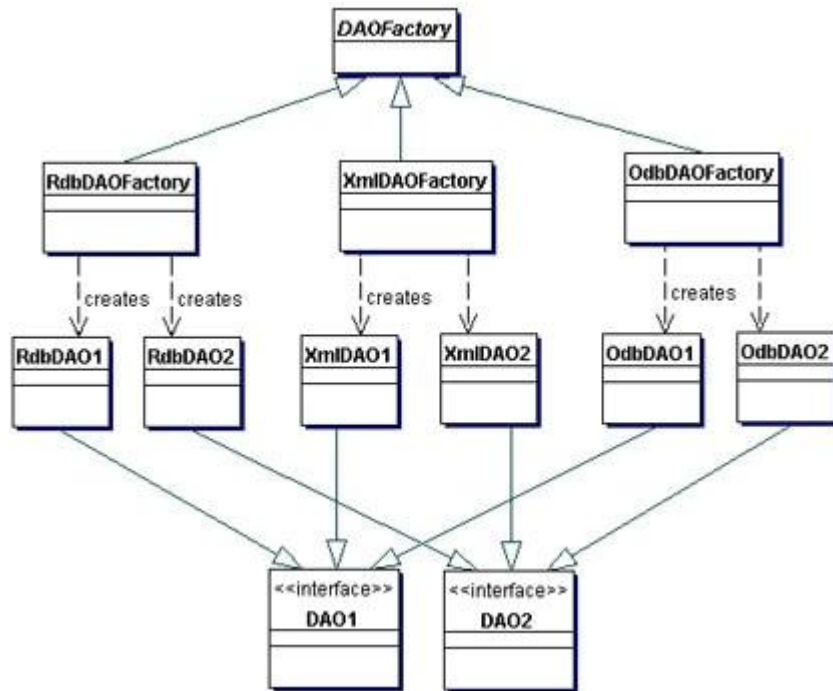


Isso representa um Objeto de Transferência usado como um portador de dados.

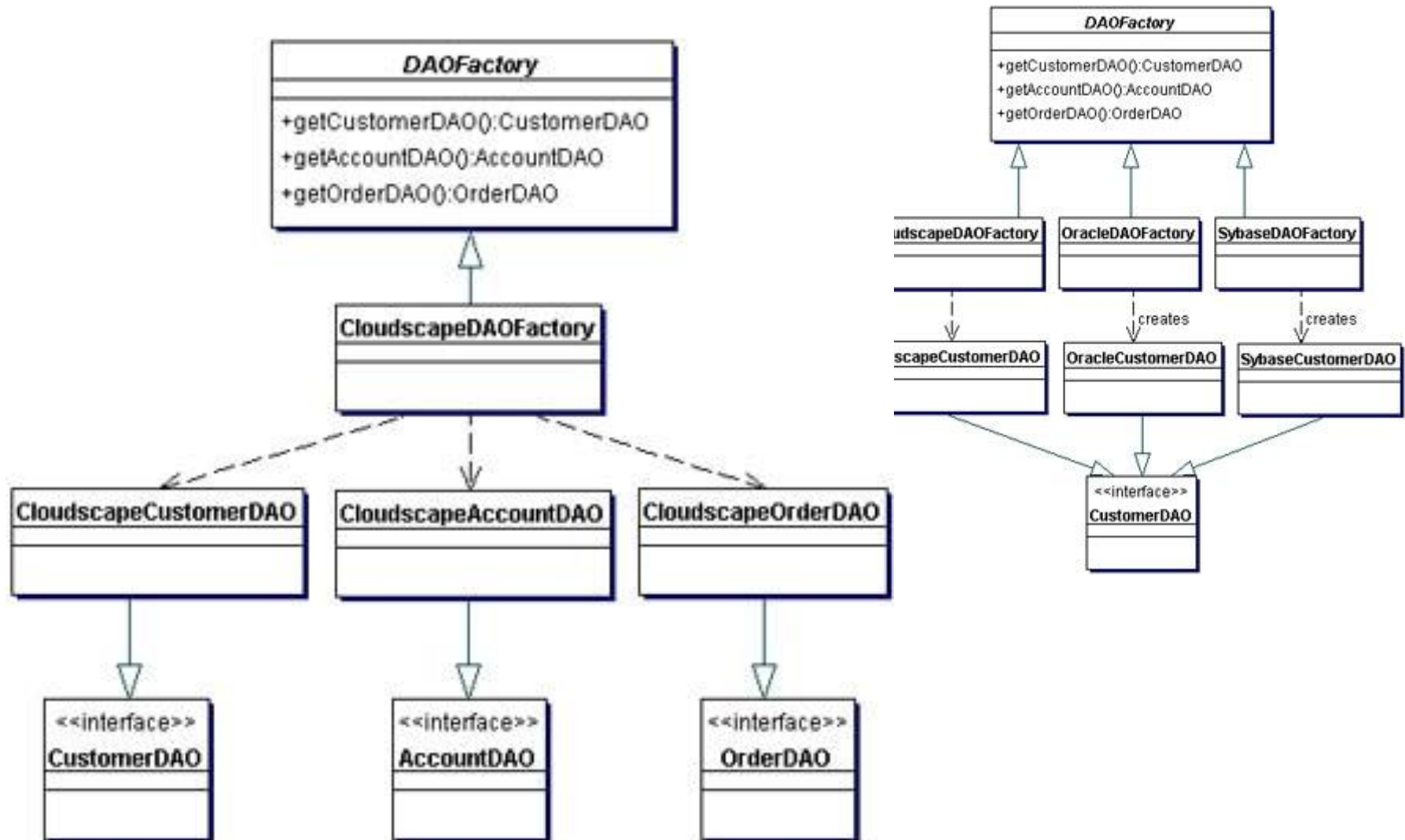
O DataAccessObject pode usar um objeto Transfer para retornar dados ao cliente.

O DataAccessObject também pode receber os dados do cliente em um objeto de transferência para atualizar os dados na fonte de dados.

Um exemplo



Um exemplo com DAOFactory



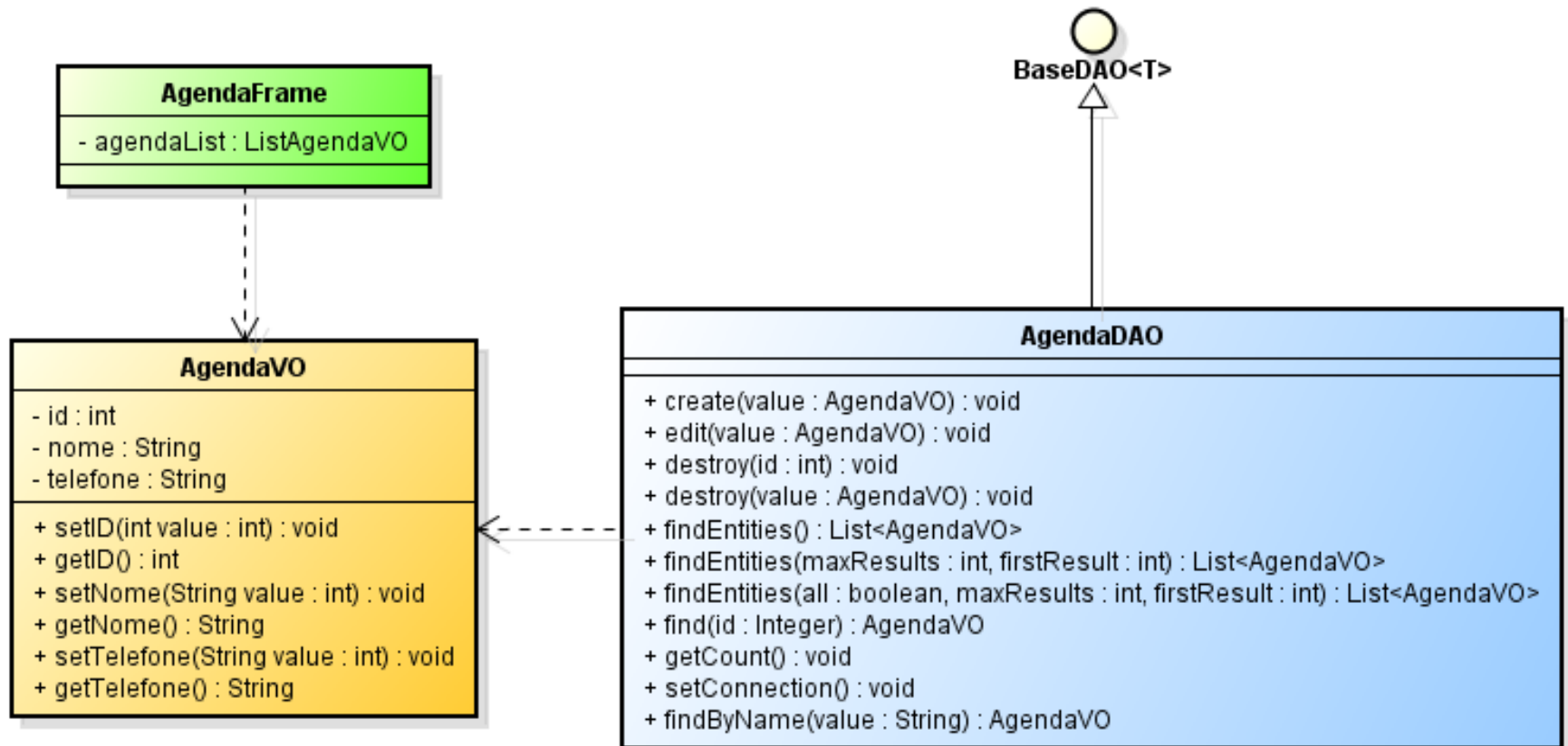
Voltando a nossa “agenda”

No exercício do cadastro da agenda:

- “Código espaguete” – tudo em uma única classe (interface, regra de negócio, entidade pulverizada, camada SQL etc
- Na versão 2: (exercício deixado em sala) – criação da classe AgendaEntradaVO (Value Object) e da classe AgendaDAO.
- Na versão 3: criação das interfaces, fabricas....

Novas classes...

okg



Referências

- Core J2EE Patterns – Data Access Object
<https://www.oracle.com/technetwork/java/dataaccessobject-138824.html>