ABBAS & TEAM

November 27, 2023
09:30 pam.

Presented To:
Sir Mukhtiar Zamin

www.daopattern..com

This event has live translations.

ABBAS & TEAM

November 27, 2023
09:30 pam.

Presented By:
Ghulam Abbas &
Tanveer Ali

www.daopattern..com

This event has live translations.

ABBAS & TEAM.

Topic:

Data Access Object Pattern

ABBAS & TEAM.

Content:

- Intent
- Participants
- Structure Diagram
- Applicability
- Consequences
- Implementation

Intent:

Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services.

Participants :

- **Data Access Object Interface**
- **Data Access Object concrete class**
- **Model Object or Value Object**

Cont...

- **Data Access Object Interface**

  This interface defines the standard operations to be performed on a model object(s).

- **Data Access Object concrete class**

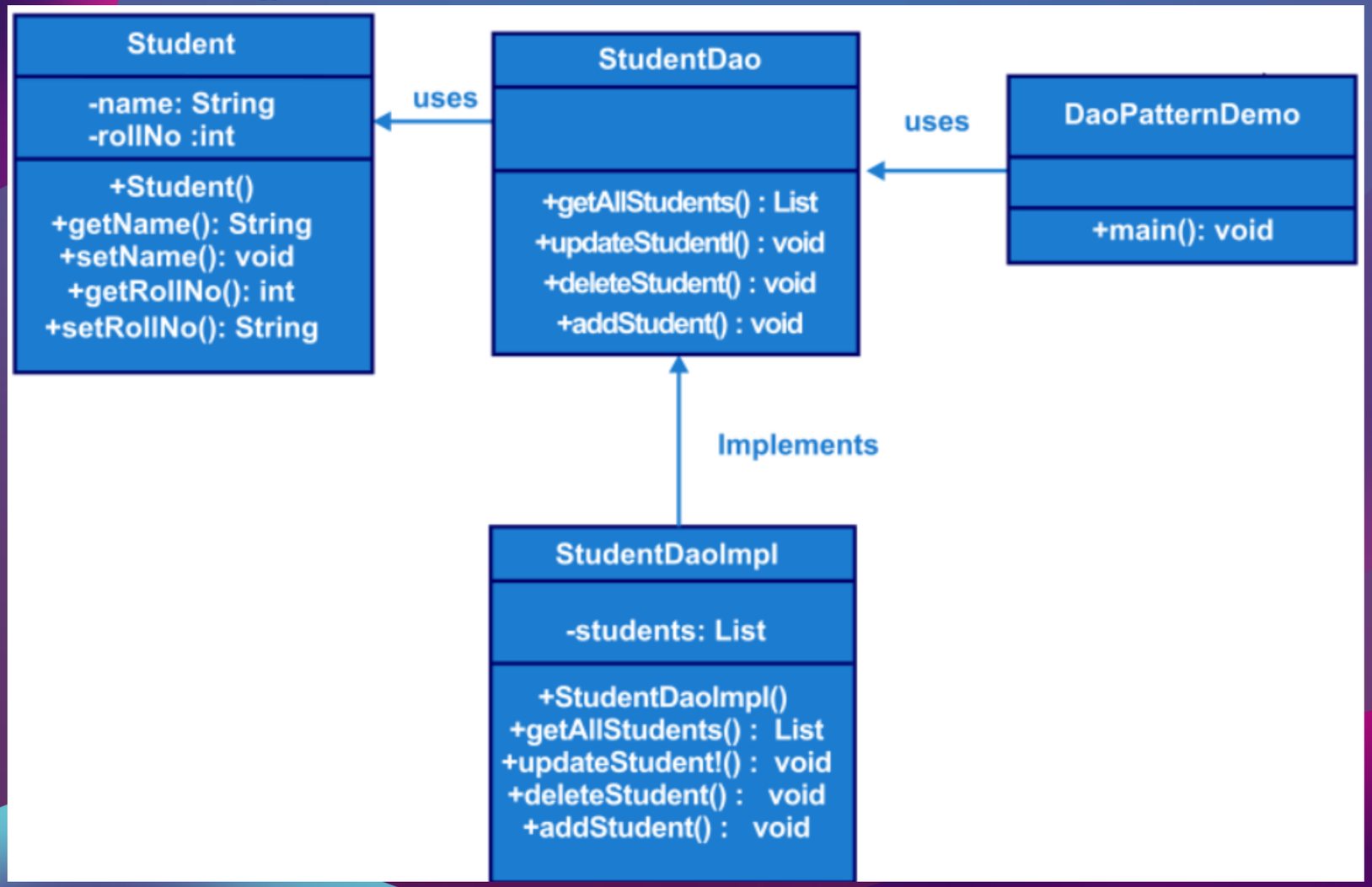
  This class implements above interface. This class is responsible to get data from a data source which can be database / xml or any other storage mechanism.

- **Model Object or Value Object**

  This object is simple POJO containing get/set methods to store data retrieved using DAO class.

November 27, 2023
09:30 pam.

Structure Diagram:



ABBAS & TEAM.

November 27, 2023
09:30 pam.

Applicability

- **Multiple Data Sources:**

If our application needs to interact with multiple data sources, such as different databases or external APIs, the DAO pattern can provide a consistent interface for data access. This helps in abstracting away the details of the underlying data storage mechanisms.

- **Code Reusability**

The DAO pattern promotes code reusability by encapsulating the data access logic into a separate layer. This makes it easier to reuse the same data access logic across different parts of the application or even in different projects.

- **Database Abstraction**

When you want to abstract the details of a specific database system (e.g., MySQL, PostgreSQL, SQLite), the DAO pattern allows you to encapsulate the database-specific code within the DAO implementation. This makes it easier to switch to a different database system without affecting the rest of the application.

ABBAS & TEAM.

November 27, 2023
09:30 pam.

Applicability

Testability

DAOs can improve the testability of your application. By having a separate data access layer, you can easily mock or substitute the data access logic during unit testing, allowing you to test the business logic independently of the actual database.

Security

If your application requires access control or security checks for data access, these concerns can be centralized within the DAO layer. This ensures that security-related logic is consistently applied across different parts of the application.

Encapsulation of Data Access Logic

The DAO pattern encapsulates the details of data access, such as SQL queries or API calls, within dedicated DAO classes. This separation of concerns helps in maintaining a clean and modular codebase.

Consequences:

Increased Complexity

Introducing the DAO pattern may add an additional layer of abstraction, potentially increasing the overall complexity of the application.

Maintenance Overhead

While DAO promotes modularity, maintaining separate DAO classes for each entity or data source may introduce overhead in terms of code maintenance.

ORM Alternatives

In some scenarios, Object-Relational Mapping (ORM) tools may provide an alternative approach to data access, and choosing between DAO and ORM may depend on project requirements and preferences.

ABBAS & TEAM.

Implementation .

ABBAS & TEAM.

Thank you!

Send us a message at gabbass731@gmail.com