

# ОБЪЕКТНО-РЕЛЯЦИОННОЕ ОТОБРАЖЕНИЕ ДАННЫХ

*Эта статья рассматривает объектно-реляционное отображение данных и примеры его реализации.*

## ВВЕДЕНИЕ

В настоящее время все больше растет потребность в приложениях, обеспечивающих использование и хранение огромного количества данных. Такие приложения используют базы данных, которые хранят в себе всю необходимую информацию, которая необходима для работы данного приложения. Однако, чтобы управлять таким количеством данных потребовалась технология, которая способна связывать программный код с базы данных, чтобы обеспечить оптимизацию приложения.

### I. ОБЪЕКТНО-РЕЛЯЦИОННОЕ ОТОБРАЖЕНИЕ

Объектно-реляционное отображение (Object-Relational Mapping) – технология программирования, позволяющая работать с данными, хранящимися в базе данных, используя объектно-ориентированное программирование. ORM является слоем абстракции между приложением и базой данных. При взаимодействии с базой данных с использованием ООП-языков вам приходится выполнять различные операции, такие как создание, чтение, обновление и удаление (CRUD) данных из базы данных. Обычно для выполнения этих операций в реляционных базах данных используется SQL. Однако ORM и инструменты ORM упрощают взаимодействие между реляционными базами данных и различными ООП-языками. Оно помогает повысить производительность разработчиков, устраняя необходимость в шаблонном коде и избежать тех или иных ошибок, которые могут появиться, в зависимости от выбранного вами языка. Таким образом, благодаря этой технологии разработчики могут использовать язык программирования, с которым им удобно работать с базой данных, вместо написания SQL-запросов или хранимых процедур.

### II. ПРИМЕРЫ РЕАЛИЗАЦИИ ORM

Существуют различные реализации объектно-реляционного отображения данных. Как правило, каждый фреймворк предназначен для конкретного языка программирования. Примеры ORM:

- Hibernate (Java): Hibernate — это мощный инструмент ORM для языка Java. Он предоставляет удобные методы для работы с базой данных, а также автоматически создает SQL-запросы на основе объектов Java. Hibernate позволяет разработ-

чикам избежать написания ручных SQL-запросов и упрощает взаимодействие с базой

- Django ORM (Python): Django ORM — это часть фреймворка Django для Python. Он обеспечивает простой и интуитивно понятный способ взаимодействия с базой данных. С помощью Django ORM можно создавать модели данных, выполнять запросы и обновлять записи без необходимости писать SQL-код вручную.
- Entity Framework (C#): Entity Framework (EF) — это ORM-инструмент для языка C#. Он позволяет разработчикам работать с данными как с объектами, а не с таблицами базы данных. EF автоматически генерирует SQL-запросы и обеспечивает удобный способ взаимодействия с базой данных.
- SQLAlchemy (Python): SQLAlchemy — это ORM-библиотека для Python. Она предоставляет гибкий и выразительный способ работы с базой данных. SQLAlchemy поддерживает различные базы данных и позволяет создавать сложные запросы с помощью высокоуровневых методов.
- Sequelize (JavaScript): Sequelize — это ORM-библиотека для Node.js и JavaScript. Она позволяет разработчикам создавать модели данных, выполнять запросы и управлять связями между таблицами. Sequelize поддерживает различные базы данных, такие как MySQL, PostgreSQL и SQLite. Эти инструменты помогают упростить взаимодействие с базой данных, сократить объем написания SQL-кода и повысить производительность разработки.

### III. ВЫВОДЫ

Использование ORM (Object-Relational Mapping) технологий имеет множество преимуществ:

- Упрощение разработки: ORM позволяет работать с базой данных в объектно-ориентированном стиле, что упрощает написание кода и снижает время разработки.
- Повышение производительности: ORM обеспечивает оптимальное выполнение запросов к базе данных, что может улучшить производительность приложения.
- Безопасность: ORM помогает предотвратить SQL-инъекции и другие уязвимости за счет использования параметризованных запросов.

- Переносимость: ORM абстрагирует код от конкретной СУБД, что упрощает перенос приложения на другую базу данных.
  - Поддержка объектно-ориентированного подхода: ORM позволяет работать с данными в виде объектов, что удобно для разработчиков, привыкших к объектно-ориентированному программированию. В целом, использование ORM технологий способствует улучшению качества и эффективности разработки программного обеспечения.
1. Kay S. Horstmann, Gary Cornell, «Java. Professional's Library» – 2020 г. – №1 – С. 120-150.
  2. Klymenko Oleksandr – «Java Reflection Api» – [Электронный ресурс] – <https://javarush.com/groups/posts/513-reflection-api-refleksija-temnaja-storona-java>
  3. Semper Ante – «Java @Annotations. What is it and how to use it?» – [Электронный ресурс] – <https://javarush.com/en/groups/posts/en.1896.java-annotations-what-is-it-and-how-to-use-it>

*Македон Вадим Александрович*, студент 3 курса факультета информационных технологий и управления БГУИРа, vadiog2@yandex.ru

*Степанова Валерия Владимировна*, студентка 3 курса факультета информационных технологий и управления БГУИРа, valerystepanova48@gmail.com

*Чаевский Спартак Владимирович*, студент 3 курса факультета информационных технологий и управления БГУИРа, spartak1469@gmail.com

*Научный руководитель: Гуревич Ольга Викторовна*, старший преподаватель кафедры вычислительных методов и программирования БГУИРа, o.gurevich@bsuir.by