**Technologie obiektowe**

**Projekt**

**Temat:** Wzorce projektowe dotyczące mapowania obiektowo – relacyjnego.

**Autor:** Bartosz Dygas

**Prowadzący:** dr inż. Mariusz Bedla

Część teoretyczna

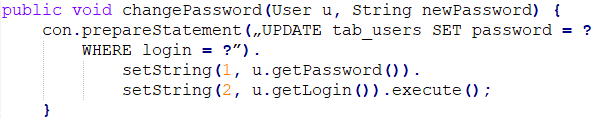
1. **Czym jest mapowanie obiektowo – relacyjne(ORM)?**

Mapowanie obiektowo – relacyjne można opisać jako sposób odwzorowania architektury obiektowego systemu informatycznego na bazę danych o charakterze relacyjnym.

Dokładniej – jest to konwertowanie tabel w bazie danych na obiekty aplikacji oraz na odwrót – konwertowanie obiektów aplikacji na tabele bazy danych.

ORM pozwala nam zastąpić poniższy kod wersją, w której nie ma konieczności bezpośrednich zmian w bazie danych z poziomu kodu.

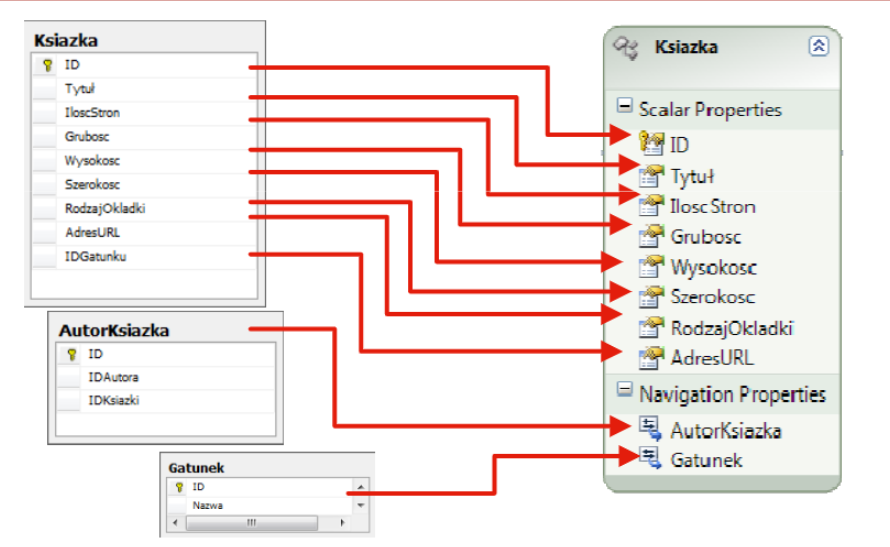
Wersja przed zastosowaniem ORM:



Wersja po zastosowaniu ORM:



ORM przedstawione graficznie:

[[1]](#footnote-1)

1. **Dlaczego stosować mapowanie obiektowo – relacyjne?**
2. **Uniezależnienie od rodzaju systemu zarządzania bazą danych(DBMS) oraz języka zapytań bazy danych.**

Dzięki wykorzystaniu odpowiedniej technologii (np. hibernate) można używać ORM bez znajomości języka SQL oraz systemu DBMS.

1. **Automatyzacja**

Dzięki wykorzystaniu ORM można zaprogramować aplikację tak, aby atrybuty obiektów automatycznie były zapisywane w bazie danych i na odwrót – tabele z baz danych były konwertowane i zapisywane do atrybutów obiektów.

1. **Redukcja ilości czasu potrzebnego na przepisywanie tabel do aplikacji lub obiektów aplikacji do tabel.**

Dzięki wykorzystaniu ORM nie tylko zmniejszamy objętość kodu aplikacji, ale też redukujemy czas potrzebny na przepisanie danych z lub do bazy danych. ORM manager zarządza optymalizacjami wydajnościowymi (pulą połączeń do bazy itp.).

1. **Uniknięcie niezgodności impedancji.**

Różnice w koncepcji języków powodują trudności techniczne w zrealizowaniu połączenia między językiem zapytań, a językiem programowania. Niezgodność impedancji to niezgodność m.in. składni, systemów typów, semantyki, poziomu abstrakcji języków. Dzięki wykorzystaniu ORM unikamy niezgodności impedancji.

1. **Jak działa mapowanie obiektowo – relacyjne?**

Mapowanie obiektowo– relacyjne polega na zdefiniowaniu odwzorowania zawartości relacyjnej bazy danych na obiekty w używanym przez nas języku programowania oraz wykonywaniu operacji na danych w bazie danych tak jak na zwykłych obiektach języka programowania.

Zasady pracy z narzędziami ORM:

1. Tworzenie modelu danych w obiektowym języku programowania
2. Tworzenie schematu bazy danych odpowiadającego modelowi obiektowego języka (jeśli taka baza danych jeszcze nie istnieje)
3. Definiowanie odwzorowania bazy danych na model relacyjny
4. Tworzenie aplikacji w oparciu o zaprojektowany wcześniej model danych.
5. W razie konieczności pobrania obiektów z bazy danych, utrwalenia nowo utworzonego obiektu lub usunięcia utrwalonego obiektu – posługujemy się odpowiednim API danego narzędzia ORM. To jest jedyne miejsce, w którym w naszej aplikacji przejmujemy się tym, że współpracujemy z jakąś bazą danych.
6. **Technologie**

Najpopularniejsze technologie wykorzystywane w mapowaniu obiektowo – relacyjnym:

1. Java EE – platforma do tworzenia aplikacji biznesowych w języku Java posiadająca technologię ORM bazującą w dużej części na Hibernate. Początkowo wykorzystywała mało dziś popularny standard EJB, który dopiero w wersji 5 został zmieniony na standard JPA.
2. Spring – platforma dedykowana do tworzenia aplikacji w języku Java. Wykorzystuje framework hibernate i został stworzony jako konkurencja dla technologii Java EE.

**Hibernate** – Framework - technologia wykorzystywana podczas tworzenia ORM w języku Java. Jego dużą zaletą jest wykorzystywany przez niego standard JPA (Java Persistence API), czyli model mapowania obiektowo-relacyjnego, który zapewnia kompatybilność zarówno z bazami SQL jak i noSQL.

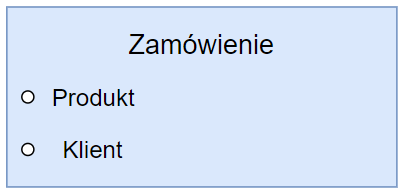
**EJB –** Jest to para interfejs + klasa. Interfejs tworzy proxy (serwer pośredniczący), do którego odwołujemy się przez interfejs.

**JPA –** Temu standardowi podlegają obiekty – encje. Są to obiekty z pewnymi ograniczeniami.

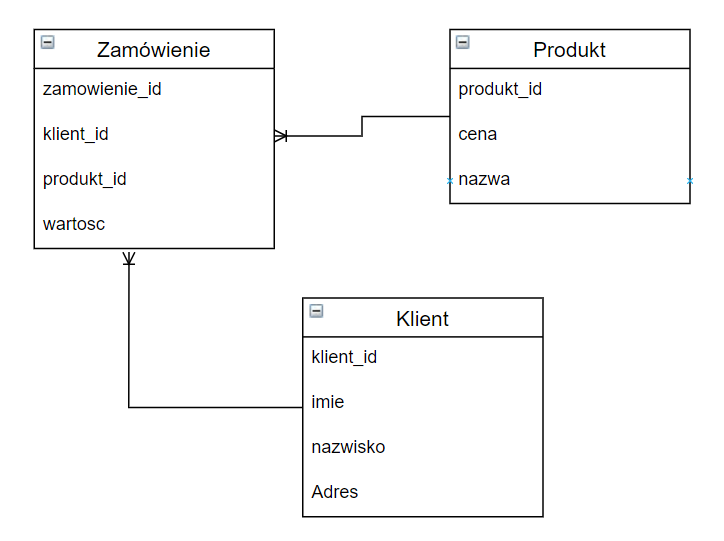
1. **Wzorce mapowania obiektowo – relacyjnego.**
2. Wzorzec Repository

Ściśle łączy się z ideą Domain Driven Design, która to z kolei zakłada, że nacisk powinien być położony na zagadnienia biznesowe, a nie na technologię. Pozwala to na modelowanie systemów informatycznych przez ekspertów, którzy znają specyfikę problemu, ale nie mają doświadczenia w projektowaniu architektury systemów informatycznych. W ramach DDD wyróżnia się „Domain objects”. Przykładowo – podczas realizacji zamówienia obiektem typu „Domain object” będzie zamówienie, które zawiera informację o zamawianym produkcie i kliencie. Natomiast na poziomie bazy danych są to 3 różne obiekty: klient, produkt, zamówienie.

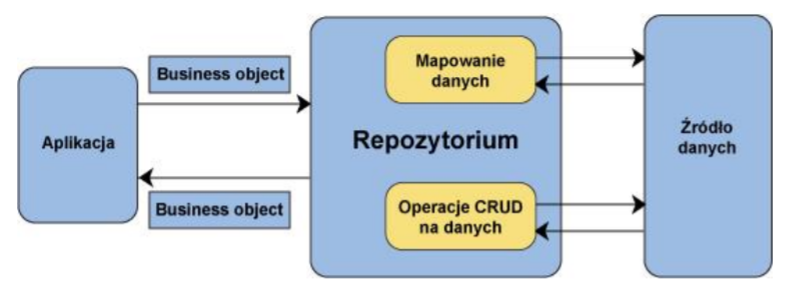
Domain object „Zamówienie”:



Obiekty w bazie danych:



Do obsługi obiektów Domain objects służa specjalne repozytoria. Dzięki nim, programiści mogą korzystać z danych zamodelowanych w taki sposób, aby odpowiadały zagadnieniom biznesowym bez konieczności tworzenia dodatkowych struktur danych. Repozytorium pośredniczy między aplikacją a warstwą mapowania danych.

[[2]](#footnote-2)

Business object na rysunku to synonim dla Domain object.

Dzięki użyciu repozytorium możliwe jest również połączenie się z kilkoma źródłami danych.

Zasady wzorca Repository:

1. Ograniczony kontekst

Wyraźnie ustalone granice pod względem organizacji zespołu, użycia w określonych częściach aplikacji i fizycznych manifestacji, takich jak bazy kodu i schematy baz danych

1. Ciągła integracja

Rozbicie systemu na małe konteksty powoduje utratę spójności i integracji. W tym celu należy wdrożyć proces częstego scalania całego kodu oraz wypracować wspólny pomysł na model, organizować spotkania wśród developerów, na których mogą opowiedzieć jaki jest ich pomysł na kontynuację projektu.

1. Mapa kontekstowa

Ograniczony kontekst sprawia, że powstaje brak ogólnego widoku projektu. Aby rozwiązać ten problem należy zdefiniować każdy model w projekcie.

1. Wzorzec ADO.Net Entity Framework.

Jest to implementacji wzorcu repository oraz Domain Driven Design.  
Głównym założeniem jest istnienie trzech modeli danych: fizycznego, logicznego oraz koncepcyjnego. Model fizyczny odnosi się do specyficznej platformy przechowywania informacji w pamięci trwałej np. sposobu zapisu danych na dyskach twardych przez MS SQL. Najważniejsze dla programistów są jednak elementy Entity Framework:

* model koncepcyjny omawiający encje i relacje w ramach systemu, który jest analizowany
* model logiczny, czyli model koncepcyjny znormalizowany dla relacyjnych baz danych, opisany tabelami i relacjami między nimi.

W procesie tworzenia oprogramowania często pomija się model koncepcyjny, który jest stosowany podczas zbierania wymagań i opisu zależności w systemach. Zespoły programistyczne od razu przystępują do modelowania zagadnienia w sposób relacyjny. Entity Framework zakłada, że model koncepcyjny jest najbardziej zbliżony do rzeczywistych wymagań stawianych wobec systemów informatycznych. Z tego powodu został wprowadzony tzw. Entity Data Model (EDM), który jest specyfikacją mapowania pomiędzy modelem koncepcyjnym a modelem logicznym

1. NHibernate

NHibernate jest to przeniesienie rozwiązania dostępnego dla języka Java na platformę .NET. Projekt jest rozwijany jako wolne oprogramowanie udostępniane na 19 licencji LGPL.

Głównym zadaniem NHibernate jest mapowanie klas języka C# na tabele bazy danych, w tym także mapowanie typów CLR (Common Language Runtime) na typy bazy danych. Jednocześnie oprogramowanie to dostarcza metod, służących do operowania na danych – zapisu, odczytu, edycji, usuwania. NHibernate jest bardzo popularnym rozwiązaniem ze względu na najbardziej rozbudowane API. Jednak przez nadmierne dodawanie funkcjonalności dokumentacja rozwiązań jest niepełna. Z tego powodu nauka tej technologii jest czasochłonna. Do poprawnej pracy wymagane jest utworzenie mapowań wskazujących odpowiednio, w jaki sposób obiekty mają być przechowywane w relacyjnej bazie danych. Dostępne są 3 metody mapowania.

* 1. Mapowanie z użyciem pliku XML
  2. Mapowanie przy użyciu atrybutów
  3. Mapowanie przy użyciu Fluent NHibernate

1. <https://silo.tips/download/mapowanie-obiektowo-relacyjne-z-wykorzystaniem-hibernate> z dnia 15.04.2021r. [↑](#footnote-ref-1)
2. <http://users.pja.edu.pl/~mtrzaska/Files/PraceMagisterskie/091026-Niegowski.pdf> z dnia 22.04.2021r. [↑](#footnote-ref-2)