

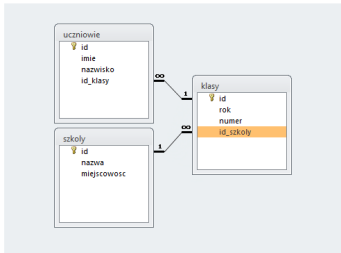
Mapowanie obiektowo-relacyjne w języku C++

Marcin Maciaszczyk, 165466
promotor dr inż. Arkadiusz Tomczyk

Politechnika Łódzka, wydział FTIMS

Łódź, 5 listopada 2013 r.

Mapowanie obiektowo-relacyjne



Relacyjne bazy danych

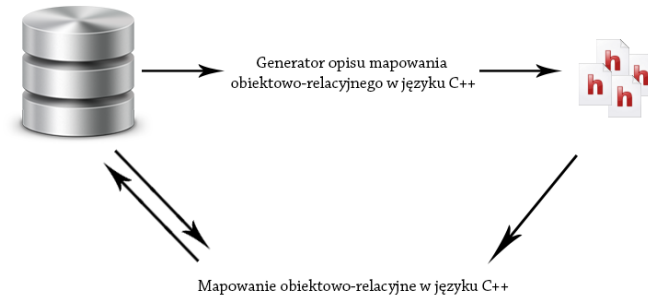
```
1 #ifndef UCZEN_H
2 #define UCZEN_H
3
4 #include<string>
5
6 using namespace std;
7
8 class uczen
9 {
10 private:
11     int id;
12     string imie;
13     string nazwisko;
14     int id_klasy;
15 public:
16     uczen();
17 };
18
19 #endif // UCZEN_H
20
```

Programowanie obiektowe

Mapowanie obiektowo-relacyjne

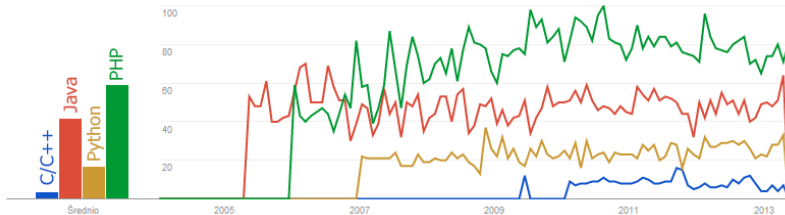
Rysunek : Mapowanie obiektowo-relacyjne (ang. Object-Relational Mapping)

Schemat aplikacji szkieletowej Qubic



Rysunek : Schemat współpracy tworzonych modułów w aplikacji szkieletowej Qubic

Uzasadnienie wyboru tematu



Rysunek : Popularność ORM w wybranych językach programowania [1]

Problematyka i zakres pracy

- Praca z zakresu mapowania obiektowo-relacyjnego oraz tworzenia aplikacji szkieletowych w języku C++
- Głównym przedmiotem pracy jest analiza istniejących aplikacji szkieletowych oraz bibliotek mapowania obiektowo-relacyjnego w języku C++ oraz zaproponowanie własnej aplikacji szkieletowej

Cele pracy

- Przeanalizowanie istniejących aplikacji szkieletowych oraz bibliotek mapowania obiektowo-relacyjnego w języku C++
- Stworzenie własnej aplikacji szkieletowej w języku C++ realizującej mapowanie obiektowo-relacyjne
- Porównanie szybkości działania przykładowej aplikacji stworzonej w oparciu o Qubic z tymi stworzonymi w oparciu o inne aplikacje szkieletowe oraz biblioteki
- Porównanie ilości kodu wymaganego aby tę samą aplikację zapisać przy użyciu Qubic oraz innych aplikacji szkieletowych i bibliotek

Przegląd literatury

Literatura dotycząca języka C++:

- Grębosz, Jerzy. Symfonia C++. Standard. Wyd. 3. Kraków, 2013. ISBN 978-83-7366-134-4 – kurs języka C++ [2]
- C++ Language Tutorial. <http://www.cplusplus.com/doc/> – dokumentacja techniczna języka C++ [3]

Litaratura dotycząca języka SQL:

- Viescas, John. Sql queries for mere mortals. 2001. ISBN 83-7279-152-X – pomoc w tworzeniu zapytań SQL-owych [4]

Przegląd literatury

Literatura dotycząca ORM:

- Barnes, Jeffrey. Object-Relational Mapping as a Persistence Mechanism for Object-Oriented Applications. Saint Paul, 2007 – teoretyczne wprowadzenie do mapowania relacyjno-obiektowego wraz z przykładami [5]
- Bauer, Christian, King, David. Hibernate w akcji. Wyd. 1. Gliwice, 2007. ISBN 978-83-246-0527-9 – opis działania ORM na przykładzie Hibernate wykorzystywanego w aplikacjach pisanych w Javie [8]

Przegląd literatury

Literatura dotycząca aplikacji szkieletowej Qt:

- Ezust, Alan and Paul. Introduction to Design Patterns in C++ with Qt4. Wyd. 1. Soughton, 2006. ISBN 978-0-13-282645-7 – wprowadzenie do Qt oraz wzorców projektowych [6]
- Qt Project Wiki. <http://qt-project.org/wiki> – dokumentacja techniczna Qt [7]

Układ pracy

- Wstęp
- Zagadnienia teoretyczne dotyczące mapowania obiektowo-relacyjnego
- Analiza istniejących aplikacji szkieletowych oraz bibliotek realizujących mapowanie obiektowo relacyjne
- Wykorzystanie aplikacji szkieletowej Qt przy tworzeniu własnej aplikacji szkieletowej
- Aplikacja szkieletowa Qubic
- Przykładowa aplikacja wykorzystująca mapowanie obiektowo-relacyjne
- Dokumentacja techniczna oraz użytkownika
- Wyniki testów
- Wnioski oraz podsumowanie

Mapowanie obiektowo-relacyjne

Mapowanie obiektowo-relacyjne

Obustronna konwersja pomiędzy danymi z tabel relacyjnej bazy danych a obiektami aplikacji [8] [9].

Aplikacja szkieletowa I

Aplikacja szkieletowa

Jedna z kategorii pakietów programistycznych, których celem jest dostarczenie szkieletu aplikacji (lub pewnej jej części), a także zbioru ogólnych funkcjonalności, które programista może rozszerzać, by dopasować je do swoich konkretnych potrzeb [10].

Aplikacja szkieletowa II

Aplikacja szkieletowa

Od biblioteki odróżnia ją kilka szczegółów:

- Odwrócenie sterowania - przepływ sterowania w aplikacji jest narzucany przez framework, nie przez programistę.
- Rozszerzalność - funkcjonalność frameworka może być rozszerzana przez programistę poprzez nadpisywanie lub dodawanie nowych elementów.
- Niemodyfikowalny kod - kod frameworka nie powinien być modyfikowany przez programistę podczas rozszerzania.

Aplikacja szkieletowa Qt

Aplikacja szkieletowa Qt

Wieloplatformowa aplikacja szkieletowa (ang. framework) dla programistów C++. Udostępnia między innymi połączenia z bazami danych, system refleksji czy też własne IDE (QtCreator) [11].

Porównanie dostępnych rozwiązań z własnym I

Nazwa rozwiązania	QxOrm	Debea	SOCI	OpenORM	Qubic
Typ rozwiązania	biblioteka	biblioteka	biblioteka	framework	framework
Obsługiwane rodzaje baz	wszystkie obsługiwane przez QtSql	MSSQL DB2 Oracle MySQL	Oracle PostgreSQL Firebird MySQL SQLite MSSQL	PostgreSQL	wszystkie obsługiwane przez QtSql
Cena	0 zł	0 zł	0 zł	0 zł	0 zł
Test wydajności ¹	?	?	?	?	?
Test czytelności ²	?	?	?	?	?

¹ Na potrzeby tego testu przy użyciu wybranych bibliotek oraz aplikacji szkieletowych realizujących ORM zapisano do bazy 1000 rekordów, następnie odczytano ich wartości, zmodyfikowano je i na końcu usunięto z bazy. Miarą efektywności jest tutaj czas wykonania cyklu operacji mierzony w ms.

² Do tego testu wykorzystano kod aplikacji napisany na potrzeby testów wydajnościowych. Miarą efektywności jest tutaj jak najmniejsza liczba linii kodu potrzebna do wykonania podstawowych operacji typu CRUD.

Rysunek : Porównanie dostępnych bibliotek oraz aplikacji szkieletowych

Porównanie dostępnych rozwiązań II

- QxOrm - pełna funkcjonalność ORM, wykorzystanie Qt, wciąż rozwijana, uciążliwa konfiguracja z poziomu kodu [12].
- Debea - podstawowa funkcjonalność ORM, kompatybilna z wieloma rodzajami baz, nierozwijana [13].
- SOCI - biblioteka ułatwiająca dostęp do danych, przy użyciu której wciąż wymagana jest konstrukcja zapytań, zupełnie podstawowa funkcjonalność, nierozwijana [14].
- OpenORM - pełna funkcjonalność ORM, open source, brak dokumentacji [15].
- Qubic - podstawowa funkcjonalność ORM, wykorzystanie Qt, łatwa konfiguracja, obszerna dokumentacja, aktualna wersja, brak konieczności własnoręcznej konstrukcji zapytań.

Użyte technologie



ORACLE®
DATABASE



Microsoft®
SQL Server®

Rysunek : Użyte technologie

Interfejs użytkownika

```
int main(int argc, char *argv[])
{
    QApplication app( argc, argv );

    Employee emp ("999999", "1953-09-02", "Marcin", "Maciaszczyk", "M", "1986-06-26");

    QbDatabase* db = QbDatabase::getInstance();

    db->connect();
    db->store(emp);
    cout << db->read(emp);

    emp.firstname = "Piotr";
    db->store(emp);
    cout << db->read(emp);

    db->remove(emp);

    return app.exec();
}
```

Rysunek : Interfejs użytkownika

Wstępne wnioski

- Qubic działa o ?% szybciej/wolniej od pozostałych aplikacji szkieletowych i bibliotek realizujących mapowanie obiektowo-relacyjne (czas w ms. – test wydajnościowy)
- Do zapisania tych samych aplikacji przy użyciu naszej biblioteki potrzeba o ?% mniej/więcej linii kodu niż w przypadku QxOrm, a także ?% mniej linii kodu niż w przypadku nie skorzystania z ORM

Perspektywy rozwoju

- Dodanie możliwości samodzielnej konstrukcji zapytań przez użytkownika
- Wielowątkowe przetwarzanie zapytań

Bibliografia I

- [1] <http://www.google.pl/trends/> – dostęp 09.11.2013.
- [2] Grębosz, Jerzy. Symfonia C++. Standard. Wyd. 3. Kraków, 2013. ISBN 978-83-7366-134-4.
- [3] <http://www.cplusplus.com/doc/> – dostęp 09.11.2013.
- [4] Viescas, John. Sql queries for mere mortals. 2001. ISBN 83-7279-152-X.
- [5] Barnes, Jeffrey. Object-Relational Mapping as a Persistence Mechanism for Object-Oriented Applications. Saint Paul, 2007.
- [6] Ezust, Alan and Paul. Introduction to Design Patterns in C++ with Qt4. Wyd. 1. Soughton, 2006. ISBN 978-0-13-282645-7.
- [7] <http://qt-project.org/wiki> – dostęp 09.11.2013.

Bibliografia II

- [8] Bauer, Christian, King, David. Hibernate w akcji. Wyd. 1. Gliwice, 2007. ISBN 978-83-246-0527-9.
- [9] <http://www.pcmag.com/> – dostęp 09.11.2013.
- [10] <http://pl.wikibooks.org/wiki/> – dostęp 09.11.2013.
- [11] <http://qt-project.org/> – dostęp 09.11.2013.
- [12] <http://www.qxorm.com/> – dostęp 09.11.2013.
- [13] <http://debea.net/trac> – dostęp 09.11.2013.
- [14] <http://soci.sourceforge.net/> – dostęp 09.11.2013.
- [15] <http://open-orm.sourceforge.net/> – dostęp 09.11.2013.