



УДК 378.147

Методика розробки механізму рефлексії для об'єктно-орієнтованого програмування в C++

Микола Бондаренко,
кандидат технічних наук, професор,
Українська інженерно-педагогічна академія, м. Харків

Антон Макаренко,
асистент,
Українська інженерно-педагогічна
академія, м. Артемівськ

*Дистанційне навчання застосовується в навчальному процесі
для самостійної роботи студентів, попереднього засвоєння
матеріалу та як спосіб закріплення пройденого матеріалу*

Підготовка бакалаврів спеціальності 6.010104.36 «Професійне навчання. Комп'ютерні технології в управлінні та навчанні» включає в себе навчальний модуль для системи дистанційного навчання за темою «Розробка механізму рефлексії для об'єктно-орієнтованого програмування в C++».

Методи дистанційного навчання — це програми, які дозволяють читати з екрана матеріал, що вивчається. Ці програми містять засоби самоперевірки якості засвоєння матеріалу, а також діалогові засоби (підказки, нагадування) тощо. Такий метод не замінить форми навчання, за яких виявляється творче сприйняття знань, здатність студентів самостійно мислити і формулювати відповіді на поставлені питання, уміння застосовувати отримані знання на практиці. Це лабораторні роботи, семінари, практичні заняття, тобто усе, що дає викладачам «зворотний зв'язок» зі студентами і допомагає їм вносити в навчальний процес необхідні корективи. Але дистанційне навчання цілком може застосовуватися в навчальному процесі

для самостійної роботи студентів, попереднього засвоєння матеріалу та як спосіб закріплення пройденого матеріалу. Створення модулів для навчання розпадається на дві основні задачі: 1) розробка програмного забезпечення; 2) розробка змістової частини — тексту навчального матеріалу, набору питань для самоперевірки і варіантів відповідей до них (інформаційного забезпечення). Робота присвячена розробці програмного та інформаційного забезпечення навчального модуля для системи дистанційного навчання за темою «Розробка механізму рефлексії для об'єктно-орієнтованого програмування в C++».

В роботі [1] розглядається оновлення змісту дисципліни «Основні напрямки розвитку в комп'ютерній галузі», де головним є об'єктно-орієнтоване програмування. Програмування виникло та розвивалося як процедурне. Процедурний підхід припускає, що основою програми є алгоритм, тобто деяка процедура обробки даних. Ускладнення завдань, розв'язуваних на ЕОМ, та збільшення їх масштабів виявили недоліки процедур-

ного підходу, пов'язані з надійністю програм, повторним використанням коду, супроводом тощо. В результаті багаторічного досвіду та аналізу цих проблем і методів на початку 90-х років минулого сторіччя з'явилася концепція об'єктно-орієнтованого програмування. Модуль для навчання складається з двох складових — програмного та інформаційного забезпечення [2]. Програмне забезпечення навчального модуля дає студенту можливість сприймати матеріал невеликими порціями (розділами) обсягом у 2–3 сторінки, тобто весь матеріал має бути добре і продумано структурований. Програмне забезпечення дає можливість відображати текст у форматovanому вигляді, що полегшує його сприйняття (колір, розмір шрифту, курсив, напівжирне креслення тощо). Об'єктно-реляційна проекція (Object-relational mapping-(ORM) — це програмна технологія, яка пов'язує реляційні бази даних з концепціями об'єктно-орієнтованого програмування шляхом створен-

ня «віртуальних об'єктних баз даних». Реалізація технології ORM є типовою для скриптових та мов програмування високого рівня, що виконуються у віртуальній машині. Ключовою вимогою до мови програмування для побудови системи об'єктно-реляційної проекції є можливість рефлексивного програмування [3]. Рефлексія дає можливість: знаходити та модифікувати конструкції коду програми як об'єкти першого класу; змінювати ім'я класів, функцій та членів класів під час виконання; розбирати та виконувати програмний код, що надходить із зовнішніх джерел, отримувати значення членів класу та викликати методи класу з їх іменами. Реалізація рефлексії не є тривіальною задачею для мов зі статичними типами даних, до яких відносяться мови C та C++. Можливості динамічної ідентифікації типів даних (RTTI), які полягають у визначенні назви класу об'єкта та визначенні належності об'єкта до певного класу у часі виконання, були значно розширені у

Таблиця 1

Аналіз підходів до реалізації механізму рефлексії в мові програмування C++

| Підхід | Переваги | Недоліки |
|---|--|---|
| Аналіз інформації для відладки. | Використання стандартного методу опису класу. Можливість отримати повну інформацію про типи даних. | Необхідність побудови програми в режимі налагодження. Залежить від компілятора. |
| Препроцесор аналізу та опису класу. | Використання стандартного методу опису класу. | Специфіка роботи компіляторів з членами класу та таблицями віртуальних функцій. Розробка аналізатора коду C++. |
| Модифікований компілятор з підтримкою рефлексії. | Відсутність додаткових етапів побудови програми. Генерація ефективного коду для засобів рефлексії. | Відтворення та адаптація програмної інфраструктури. |
| Ручне формування мета-опису класу засобами мови програмування. | Незалежність від компілятора. Проста реалізація. | Написання додаткового програмного коду, для опису метаінформації. Людський фактор. |
| Формування метаопису класу в конфігурації зовнішніх класів. | Використання стандартного методу опису класу. Можливість використання стандартних засобів опису структур даних (XML, JSON). | Додаткові часові та ресурсні видатки на отримання метаінформації. |
| Формування мета-опису класу засобами скриптової мови програмування. | Використання вбудованих можливостей скриптових мов програмування. | Часові та ресурсні витрати на запуск та роботу з інтерпретатором скриптової мови програмування. |

новому стандарті C++. На даний час існує декілька відомих реалізацій механізму рефлексії для мови програмування C++:

- реалізація C++/CLI від корпорації Microsoft;
- бібліотека Mirror, яку засновано на останньому стандарті мови програмування C++;
- Qt Framework.

В табл. 1 подано отримані результати аналізу підходів до організації рефлексії для мови програмування C++.

Метакомпіляція (англ. *metacompilation*) — це прийом програмування, який дозволяє оптимізувати програмний код алгоритму. Вона дозволяє мінімізувати використання засобів статичного поліморфізму. Метадані використовуються як основна складова не тільки для опису класу, але й для опису реляційних таблиць та їх складових. Етап метакомпіляції складається з синтаксичного, лексичного аналізу та генерації метаоб'єктного коду. У результаті метакомпіляції мають бути сформовані:

- опис класу об'єкта віртуальної об'єктно-орієнтованої бази даних;
- допоміжний метаоб'єкт класу C++;
- вхідні дані для компіляції;
- опис реляційної таблиці.

Методично СКБД MySQL описується за наступною схемою:

```
CREATE [TEMPORARY] TABLE
[IF NOT EXISTS] tbl_name
[(create_definition,...)] [table_
options] [select_statement]
create_definition:
col_name type [NOT NULL |
NULL] [DEFAULT default_value]
[AUTO_INCREMENT] PRIMARY KEY]
[reference_definition]
| PRIMARY KEY (index_col_
name,...)
| KEY [index_name] (index_
col_name,...)
| INDEX [index_name] (index_
col_name,...)
| UNIQUE [INDEX] [index_
name] (index_col_name,...)
```

```
| FULLTEXT [INDEX] [index_
name] (index_col_name,...)
| [CONSTRAINT symbol] FOREIGN
KEY [index_name] (index_col_
name,...) [reference_definition]
или CHECK (expr)
```

На основі аналізу структурних схем опису реляційних таблиць в СКБД, слід визначити необхідний перелік базових характеристик, які потрібні для опису полів метаоб'єктної таблиці:

1. Ім'я поля реляційної таблиці.
2. Тип даних, що зберігаються в полі.
3. Кількість знаків, що зберігаються в полі.
4. Можливість заборони пустих полів.
5. Значення, що присвоюється полю таблиці за замовченням.
6. Визначення первинного ключа таблиці.

Загальними параметрами для поля класу і поля реляційної таблиці з встановленого переліку є ім'я та тип даних.

На основі співвідношення типів даних, структури опису поля таблиці пропонується наступна форма опису поля метаоб'єктної таблиці:

```
ТИП_ДАНИХ [ (КІЛЬКІСТЬ_ЗНАКІВ
[, КІЛЬКІСТЬ_ЗНАКІВ] ) ]
ІМ'Я_ПОЛЯ
[NOT NULL]
[DEFAULT]
```

Для відокремлення полів метаоб'єктної таблиці необхідно використовувати наступний маркер доступу — *property*.

```
CREATE TABLE Person (
id INT AUTO_INCREMENT PRIMARY
KEY,
name VARCHAR(20) NOT NULL,
surname VARCHAR(30) NOT NULL
)
class Person{
properties:
VARCHAR(20) name NOT NUL;
VARCHAR(30) name NOT NUL;
};
```

Поле id у даному випадку генерується автоматично.

В роботі отримано такі основні результати.

1. Засобами середовища Delphi визначена методика розробки модуля для системи дистанційного навчання за темою «Методика розробки механізму рефлексії для об'єктно-орієнтованого програмування в C++».

2. Проведено перевірку роботи цього модуля та аналіз загальних вимог, висунутих до навчального модуля.

3. Розроблено відповідне програмне та інформаційне забезпечення для навчального модуля.

4. Показано приклади роботи програми, що демонструють коректність її функціонування.

Література

1. Бондаренко, М. А. Основи інформаційних технологій та програмування. — Харків : ФОП Павленко О.Г., 2010. — 536 с.

2. Бондаренко, М. А. Програмування на Object Pascal в середовищі Delphi. — Харків : Бізнес Інформ, 2008. — 704 с.

3. Coleman, G., Verbruggen R. A Quality Software Process for Rapid Application Development // Software Quality Control. — 1998. — Vol. 7, № 2. — P. 107–122.

20.09.2015.