

УДК 621.396

П.С. Венгерський*, О.В. Єфремов*, Б.М. Стрихалюк#

*Львівський Національний університет ім. І. Я. Франка,
кафедра інформаційних систем,

#Національний університет "Львівська політехніка", кафедра телекомунікацій

ЗАСТОСУВАННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПІДХОДУ ДЛЯ ОПИСУ АЛГОРИТМІВ РОЗВ'ЯЗУВАННЯ КРАЙОВИХ ЗАДАЧ МЕТОДОМ СКІНЧЕННИХ ЕЛЕМЕНТІВ

© Венгерський П.С., Єфремов О.В., Стрихалюк Б.М., 2002

У цій статті наведено основні концепції об'єктної моделі "дані-операції-класи операцій".

This paper introduce basic concepts of "data-tools-set of tools" object model.

1. Вступ

За останні роки в розвитку інформаційних технологій набуло широкого застосування об'єктно-орієнтоване програмування, тому всі поширені мови програмування підтримують цю концепцію. Завдяки цьому розробка програмного забезпечення почала відбуватися швидше та якісніше.

Як наслідок, з'явилося багато об'єктно-орієнтованих бібліотек класів для розв'язування різних типів задач. Проте більшість з них є орієнтована на роботу з базами даних, комп'ютерною графікою, засобами мережевого зв'язку тощо.

Для розв'язування задач математичного моделювання необхідна наявність бібліотеки, яка б оперувала такими поняттями, як вираз, функція, рівняння.

Ця бібліотека має володіти такими властивостями:

- робота з виразами в аналітичній формі;
- універсальна функція розбору виразів;
- можливість обчислення інтегралів та диференціалів в аналітичному вигляді;
- стандартизований формат опису виразів у зовнішніх файлах та можливість переносу інформації між різними програмами.

Використовуючи таку бібліотеку як основу, можна реалізувати різні математичні методи. Завдяки тому, що методи будуть реалізовані з використанням спільної базової бібліотеки, легко забезпечити сумісність між ними. Крім того, ми отримуємо переваги аналітичного представлення виразів.

Подібною функціональністю володіють такі математичні пакети, як Reduce, Mathematica, MathCad. Але вони є пакетами, а не бібліотеками класів. Вони дають можливість користуватись лише вбудованими в ці пакети функціями, можливості їх внутрішніх мов є обмеженими. Використання ж бібліотек класів, ніяк не обмежує потенційні можливості майбутніх програм. Такі бібліотеки можна легко розширити чи модифікувати для забезпечення специфічної функціональності.

2. Концепція моделі "дані-операції-класи операцій"

Основними сутностями системи є такі об'єкти:

- дані;
- операції;

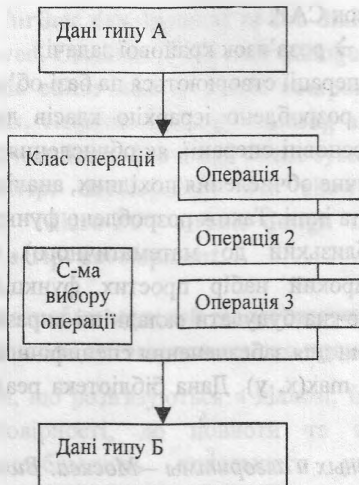
- класи операцій;

До даних ми зараховуємо будь-який об'єкт, який можна відокремити від системи та використати іншою системою. Дані є статичними, їх не можна змінити. Змінюючи будь-які дані, отримуємо нові, але не знищуємо старих. Дані повинні володіти властивістю серіалізації – це спростить механізм імпорту/експорту даних між різними системами. Приклад: вхідні дані, результати роботи програми, проміжні результати, графіки, таблиці, рівняння тощо.

До операцій ми зараховуємо об'єкти, що можуть виконувати певні функції. Об'єкти можуть отримувати деякі дані у вигляді параметрів операцій та повертати нові дані, як результат своєї роботи. Кожна операція містить певний алгоритм. Операції можуть мати властивості даних. Приклад: побудова графіка функції, розв'язування СЛАР методом Гаусса, алгоритм розв'язування крайової задачі.

Класи операцій – це набори операцій, що отримують на вході однотипні параметри та повертають однотипні результати. Завдання класів операцій – вибрати найкращу з можливих операцій в певній конкретній ситуації. Також класи операцій можуть динамічно змінювати множину доступних операцій. Приклад: розв'язування СЛАР.

Щоб застосувати якусь операцію, вона повинна належати до одного з класів операцій, також операція може належати до кількох класів одночасно.



Використовуючи дану модель, ми створюємо необхідні класи “даних”, класи “операцій” та класи “класів операцій”. Далі, використовуючи об'єкти створених класів, ми описуємо алгоритм для розв'язання поставленої задачі. При описі алгоритму ми не фіксуємо конкретних операцій, що будуть застосовуватись до наших даних, а лише визначаємо до якого класу дана операція має належати. Такий підхід дає можливість описувати алгоритм в загальному вигляді, не звертаючи увагу на особливості конкретних задач. Пізніше можна буде створити відповідну операцію для кожного конкретного типу даних. Це дасть змогу легкого розширити можливості створеного програмного забезпечення лише за рахунок додавання/заміни операцій. Після того, як було розроблено алгоритм розв'язання деякої задачі, ми можемо створити відповідну “операцію” і “клас операцій” для подальшого його використання при розв'язання складніших задач.

3. Опис алгоритму розв'язування крайових задач з використанням методу скінченних елементів

Багато задач математичної фізики часто зводяться до розв'язування крайових задач. Для знаходження числового розв'язку таких задач широко використовується метод скінченних елементів.

В даній статті запропоновано опис цього методу, використовуючи концепцію об'єктної моделі "дані-операції-класи операцій".

Метод скінченних елементів

Вхідні дані:

- опис крайової задачі;
- опис області;

Вихідні дані:

- розв'язок крайової задачі;

Операції, потрібні для реалізації МСЕ

1. крайова задача \rightarrow варіаційна задача (ВЗ);
2. варіаційна задача \rightarrow базисні функції (БФ);
3. опис області \rightarrow розбиття області (РО);
4. (ВЗ) + (БФ) + (РО) \rightarrow САР;
5. САР \rightarrow розв'язок САР;
6. розв'язок САР \rightarrow розв'язок крайової задачі;

Всі перелічені дані та операції створюються на базі об'єктів для представлення виразів в аналітичній формі. Було розроблено ієрархію класів для представлення аналітичних функцій. Є реалізовані такі основні операції, як обчислення значення виразу, перетворення (спрощення) виразів, аналітичне обчислення похідних, аналітичне обчислення інтегралу для деякої підмножини функцій та інші. Також розроблено функцію розбору виразів, при цьому забезпечується простий (близький до математичного) опис виразів та рівнянь. В розробленій ієрархії є широкий набір простих функцій. Використовуючи механізм суперпозиції, на їх основі можна будувати складніші вирази. Ієрархію класів можна легко розширити новими функціями для забезпечення специфічної функціональності. Наприклад, додавання такої функції, як $\max(x, y)$. Дана бібліотека реалізована мовою Object Pascal в середовищі Delphi 5.0.

1. Ахо, Структуры данных и алгоритмы – Москва: Вильямс, 2000 – 384 с. 2. Бумфрей XML. Новые перспективы WWW. Расширенный язык разметки, – Москва: ДМК, 2000. 3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами на C++ – Москва: Бино, 1999. 4. Коннор Дж., Бреббиа К., Метод конечных элементов в механике жидкости. Перевод с английского – Ленинград: Судостроение, 1979. 5. Шайдунов В. В., Многосеточные методы конечных элементов – Москва: Наука, 1989. 6. Шинкаренко Г. А., Проекційно-сіткові методи розв'язування початково-крайових задач – Київ: УМК ВО, 1991. 7. Эбнер М. Delphi 5. Руководство разработчика – Киев: BHV, 2000. 8. D. Eyheramendy – Finite Elements in Analysis and Design, 2000. 9. R. Foerch, Un environnement orienté object pour la modélisation numérique des structures, Ph. D. Thesis, Ecole Nationale Supérieure des Mines de Paris, 1996. 10. B.W.R. Forde, R.O. Foschi, S.F. Stiemer, Object-oriented finite element analyses, *Comput. & Struct.* 34 (1990) 355-374. 11. J. Besson, R. Foerch, Large scale Object-oriented finite element code desig, *Computation Methods Appl. Mech. Eng.* 142 (1997) 165-187.