

СТРАТЕГІЯ, ЗМІСТ ТА НОВІ ТЕХНОЛОГІЇ ПІДГОТОВКИ СПЕЦІАЛІСТІВ З ВИЩОЮ ТЕХНІЧНОЮ ОСВІТОЮ

УДК: 370+000

І. П. Дончев;

Е. С. Тодорова, докт., доц.

ОПЕРАЦІЇ З ОБ'ЄКТАМИ Й КОМУНІКАЦІЯ МІЖ ОБ'ЄКТАМИ У НАВЧАННІ ОБ'ЄКТНО-ОРІЄНТОВАНОМУ ПРОГРАМУВАННЮ

Для успішного застосування об'єктно-орієнтованого підходу в програмуванні дуже важливо, щоб студенти розуміли динаміку процесів у програмі й набули вміння ефективно моделювати комунікацію й взаємодію між об'єктами. Формування таких умінь є метою об'єктно-орієнтованого аналізу й дизайну (ООАД) і об'єктно-орієнтованого програмування (ООП).

Проаналізовано й сформульовано деякі недоліки навчальної літератури в області програмування, що стосуються операцій з об'єктами й комунікації між об'єктами. Запропоновано систему базових понять, необхідних для курсу з ООП; запропоновано класифікацію операцій з об'єктами відповідно до цілей курсу, а також наведені деякі рекомендації для викладачів.

Вступ

В процесі дослідження навчальної літератури можна помітити несприятливу тенденцію концентрувати навчання ООП на вивченні синтаксичних конструкцій для конкретної об'єктно-орієнтованої мови. Ще у вступних темах докладно коментуються залежні від конкретної мови особливості, які часто стосуються не принципів об'єктно-орієнтованого підходу, а швидше ефективності програмної реалізації. Ці особливості, безумовно, є важливими, але якщо метою курсу є формування у студентів об'єктно-орієнтованого мислення, то в першу чергу треба вивести загальні для парадигми концепції й механізми.

Іншою такою тенденцією є акцентування на даних, описаних у класі та їхній структурі, але менше уваги приділяється комунікації між об'єктами й операціям з ними. Класи й об'єкти розглядаються ймовірно як додаткова мовна конструкція, що доповнює процедурне програмування. Для об'єктно-орієнтованого підходу важливою є декомпозиція системи класів і означення її динаміки в термінах передачі повідомлень між спільно працюючими об'єктами цих класів. «Способом фокусування на передачу повідомлень вже в перший день навчання студенти можуть дійти до програмної моделі, в якій не алгоритми, а відношення й комунікації між об'єктами посідають центральне місце в проектуванні програми» [1].

Основні поняття, пов'язані з комунікацією між об'єктами

В ООП програму потрібно розглядати як систему взаємодіючих між собою об'єктів, які здійснюють комунікацію між собою способом повідомлень. Об'єкти реагують на отримані повідомлення, змінюючи свій стан завдяки викликаним методам, що зі свого боку може призвести до нових повідомлень, спрямованих до інших об'єктів. Саме ця комунікація на думку Алана Кей є другим із шести принципів першої успішної мови для ООП — Smalltalk:

«Об'єкти зв'язуються між собою способом відправлення й одержання повідомлень» [2, с. 534].

Таким чином, графічно, робота однієї об'єктно-орієнтованої програми виглядає так, як наведено на рис. 1. Стрілки вказують напрямок передачі повідомлень.

«Сама головна характеристика класу в тому, які повідомлення йому можна відправляти» [3, с. 31]. Ось чому особливо суттєвою є проблема з'ясування змісту понять метод і повідомлення, зв'язку між ними й механізму передачі повідомлень.

У книгах з програмування, що розглядають як процедурні, так і об'єктно-орієнтовані аспекти, наприклад [4, 5, 6, 7, 8] найчастіше за все передача повідомлень не коментується. Мова йде тільки про роль методів (частіше названих члени-функції) маніпулювання даними в об'єкті. Дуже часто в прикладах міститься один єдиний клас і комунікація між об'єктами не обговорюється. Напевно, причина полягає в тому, що такі підручники використовують гібридну мову (найчастіше C++) і особливу увагу приділяють алгоритмічній і процедурній можливостям, та структурам даних і їх ефективній програмній реалізації. Крім того, сама мова сприяє повному використанню переваг обох парадигм.

У книгах, виданих тільки з метою вивчення ООП, але потребуючих знання процедурних особливостей мови, також не приділяється уваги передачі повідомлень, але йдеться про те, що «комунікація із представниками класу здійснюється тільки через члени-функції» [9, с. 35] і «активування методу об'єкта можна порівняти з відправленням повідомлення до об'єкта про те, які дії він повинен виконати» [9, с. 45].

У чистих об'єктно-орієнтованих мовах використовується термін «повідомлення». Кажуть, що даний об'єкт відправляє повідомлення іншому, а між тим у гібридних мовах, як наприклад C++, частіше зустрічаємо пояснення того ж самого механізму, як викликання члена-функції об'єкта.

Найбільшу увагу механізму передачі повідомлень, зв'язкам і комунікаціям між об'єктами приділено в книгах з ООАП [10, 11, 12]. В них стверджують, що «насправді, повідомлення є операцією, яку один об'єкт виконує для іншого, незважаючи на різні керувальні механізми. В наших цілях терміни операція й повідомлення є тими ж самими» [12, с. 84].

З огляду на особливості навчального курсу ООП, пропонуємо такі означення основних понять, пов'язаних із взаємодією між об'єктами:

Означення 1. Метод: член-функція класу. Визначає що саме можуть робити об'єкти цього класу (визначає операцію над об'єктами класу). Методи кожного класу визначають спосіб маніпулювання даними цього класу, а потім і його функціональність.

Означення 2. Повідомлення: подача заявки об'єкта на виконання методу. Містить ім'я методу й вартості його аргументів. Виклик методу можна розглядати як реакцію на отримане повідомлення.

Означення 3. Операція: «послуга, яку клас може надавати своїм клієнтам» [12, с. 86].

В об'єктно-орієнтованих мовах операції, що виконуються з об'єктами, як правило, є частиною означення класу, який є типом об'єктів (методи). Для деяких мов (наприклад Object Pascal, C++, CLOS і Ada) допустима реалізація операцій із зовнішніми для класу функціями (звичай дружніми), але це порушує вимогу інкапсуляції даних і не рекомендується, особливо на початкових етапах навчання ООП. У [13] механізм дружніх функцій названо «вбивцею інкапсуляції». Звичайно, є випадки, коли реалізація операцій із зовнішньою функцією насправді необхідна для ефективності й елегантності програми. Такими є операції над об'єктами різних класів або непрості операції над об'єктами одного й того самого класу.

Таким чином, дійшли висновку, що всі методи — це операції, але не всі операції є методами.

Слід зазначити, що операцію можна здійснити й іншим способом — статичним методом (так званим методом класу). Статичні методи асоційовані в класі й вони не мають жодного відношення до його об'єктів.

Означення 4. Стан об'єкта: поточні вартості його членів-даних.

Означення 5. Поведінка об'єкта: те, як об'єкт реагує на отримані повідомлення, як змінюється його стан, які повідомлення він передає.

Означення 6. Ідентичність об'єкта: це те, що відрізняє об'єкт від інших об'єктів.

Означення 7. Інтерфейс класу: частина класу, що є «зримою» для зовнішнього середовища. Саме завдяки їй об'єкти здійснюють комунікацію із зовнішнім для них середовищем.

Поняття інтерфейсу не є в достатньо широким, щоб охопити всю взаємодію об'єкта із зовніш-

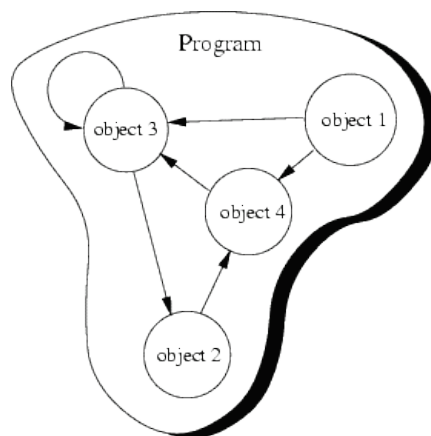


Рис. 1. Програма, що складається з чотирьох об'єктів

нім середовищем. Тому Буч використовує поняття «протокол», що містить крім методів і всі операції, пов'язані з об'єктом.

Означення 8. *Протокол* об'єкта: «Сукупність всіх методів і вільних процедур, що відносяться до конкретного об'єкта, утворює протокол цього об'єкта. Протокол, таким чином, визначає поведінку об'єкта, що охоплює всі його статичні й динамічні аспекти.» [12, с. 88].

Класифікація операцій

У більш старих підручниках з програмування й структур даних немає класифікації основних операцій з абстрактними типами даних. З метою вивчення ООП така класифікація є дуже корисною. Головна мета — акцентування уваги на загальному в операціях для різних класів.

Аналізуючи запропоновані у [14, 15, 16, 8, 12] класифікації, автори постали перед проблемою, що часто зустрічається в комп'ютерній літературі — з термінологією. З одного боку різні автори вкладають різний зміст у терміни (це значною мірою залежить від підручника — процедурного програмування, структур даних, ООП або ООАП), а з іншого — переклад термінів на різні мови теж не є однозначним.

З урахуванням мети курсу ООП, пропонуємо класифікацію операцій з об'єктами, показану на рис. 2.



Рис. 2. Класифікація видів операцій

Конструктори — це операції, які призначені для створення об'єктів, а *деструктори* — для їхнього знищення.

Селектори призначені для отримання поточних вартостей членів-даних об'єкта, причому ці дані змінити неможливо.

Модифікатори (називані ще мутаторами) призначені для зміни вартостей членів-даних об'єкта.

Предикати можна розглядати як спеціальний вид селекторів, уособлених в окремій категорії, тому що, прочитавши вартості членів-даних, лише вони повертають булевий результат (true або false).

Ітератори використовують лише для об'єктів контейнерних класів (реалізації списку, черги, графу, стека тощо). Вони забезпечують доступ до даних у чітко визначеній послідовності.

Складені операції — це комбінація декількох операцій, наприклад, сортування контейнерного класу або обробка даних класу, причому необхідне використання селекторів, модифікаторів тощо.

Висновки та рекомендації

1. Для правильного засвоєння підходу рекомендуємо у вступній частині навчального курсу ООП обмежити реалізацію операцій способом статичних методів і дружніх функцій. Тим більше, що в чистих об'єктно-орієнтованих мовах (наприклад Smalltalk) така реалізація неприпустима.

2. Вивчаючи динамічні аспекти об'єктів, треба з'ясувати механізм, завдяки якому методи отримують доступ до об'єкта, тобто, додатковий прихований параметр — покажчик на об'єкт, для якого вони викликані. У різних мовах цей покажчик позначається як *this*, *self* або *Me*.

3. Для повного засвоєння об'єктно-орієнтованого підходу необхідно розглядати об'єкти з пог-

ляду їх стану, поведінки й ідентичності.

4. У навчальних курсах ООП важко знайти приклади, що використовують складні об'єкти з нетривіальною поведінкою. Проте, в промисловості часто необхідно використовувати саме такі об'єкти, над якими працює безліч програмістів. Через обмежену кількість лабораторних робіт студентам слід пропонувати більш складні приклади, як проекти у напівготовому вигляді, до яких вони повинні додати функціональність.

5. Для формування навичок роботи в команді студентам треба давати у розробку курсові проекти більших обсягів.

СПИСОК ЛІТЕРАТУРИ

1. Bergin J. Teaching polymorphism early / Bergin J., Wallingford E., Caspersen M., Goldweber M., Kolling M. // Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education. — Caparica, Portugal : ACM, 2005, P. 342 — 343.
2. Kay A. C. The Early History of Smalltalk, in Bergin T. J. & Gibson R. G., History of Programming Languages — II / A. C. Kay. — ACM Press, 1996. P. 511—598.
3. Eckel B. Thinking in Java / B. Eckel.— [4 edition] : Prentice Hall. — 2006. — 1150 p.
4. Liberty J. Sams Teach Yourself C++ in 21 Days. / Liberty J., Jones B. — [5th edition]. — Sams, 2004, 936 p.
5. Gosling J., Bill Joy, Guy Steele and Gilad Bracha. The Java™ Language Specification. — Prentice Hall PTR. — [3 edition]. — June 24, 2005. — 688 p.
6. Hekmat Sh. C++ Essentials. PragSoft. — 2005. — 311 p.
7. Kalev D. ANSI/ISO C++ Professional Programmer's Handbook / D. Kalev. — Que, 1999. — 356 p.
8. Тодорова М. Програмиране на C++, Първа и Втора част. / М. Тодорова. — СИЕЛА, София : 2002. — 482 стр.
9. Богданов Д. Обектно-ориентирано програмиране със C++ / Д. Богданов. : Техника, 1994. — 239 стр.
10. Rumbaugh J. The Unified Modeling Language Reference Manual. [Second Edition] / Rumbaugh J., Jacobson I., Booch G. — Addison Wesley, 2004. — 752 p.
11. Hortsman C. Object-Oriented Design and Patterns / C. Hortsman. — Wiley, 2004, 472 p.
12. Booch G. Object-Oriented Analysis And Design With Applications. [second edition] / G. Booch. Addison-Wesley, 1998. — 543 p.
13. Zhu H. Methodology First and Language Second: A Way to Teach Object-Oriented Programming / H. Zhu, M. Zhou // Companion of the 18th annual ACM SIGPLAN conference on OOPSLA, Anaheim, CA, USA, 2003. — P.: 140—147.
14. Lippman St. C++ Primer / Lippman St., Lajoie J. — Addison Wesley, 1998. — 1264 p.
15. Азълв П. Информатика за 10-ти клас Профилирана подготовка / П. Азълв, Ф. Златарова, М. Тодорова. — София : Просвета, 2003. — 352 стр.
16. Deitel H. M. C How to Program / Deitel H. M., Deitel P. J. — Prentice Hall. — [3 edition]. — August 16, 2000. — 1253 p.

Рекомендована кафедрою програмного забезпечення

Надійшла до редакції 8.09.08
Рекомендована до друку 20.10.08

Дончев Ивайло Петрович — викладач; **Тодорова Емілія Стойчовна** — завідувач кафедри.

Кафедра інформаційних технологій Великотирновського університету ім. Св. Кирила і Мефодія, Болгарія, м. Велике Тирново