Семеріков С.О.
Криворізький державний педагогічний університет

ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД ЯК ЗАСІБ АКТИВІЗАЦІЇ ПІЗНАВАЛЬНОЇ ДІЯЛЬНОСТІ

"Наука про те, як замість лише бистродії машини використовувати чисельні методи та бібліотечні програми, переживає період дитинства і є однією з найважливіших галузей дослідження у майбутньому" [8, с. 198]. Ці слова Р.В. Хеммінга, сказані майже 40 років тому, і нині не втрачають своє актуальності. Тенденція до створення та систематичного використання математичних бібліотек, що виникла на початку 60-х рр., з кінця 80-х рр. стає домінуючою. Проте розвиток чисельних методів обумовлюється сьогодні також розвитком засобів обчислювальної техніки та технології програмування. Так, поява паралельних ЕОМ породила новий клас чисельних методів, орієнтованих на паралельні обчислення, а розвиток *об'єктно-орієнтованого програмування* (ООП) – новий напрямок, який сьогодні є провідним у чисельних методах: *об'єктні обчислення*.

Природну спільність принципів ООП та методологічних основ обчислювальної математики було досліджено у 80-90-х рр. в роботах Й. Арндта, В.П. Іваннікова, Р. Позо, О.П. Поліщука, В.А. Семенова, К.Є. Ширяєвої, Б. Хайбле та інших. Проте звертає на себе увагу той факт, що, незважаючи на широке застосування об'єктного підходу в практиці обчислень, застосування ООП до викладання чисельних методів у вищій школі і досі залишається недослідженою проблемою. Відсутність методики викладання чисельних методів у об'єктній методології привела до необхідності створення відповідного навчально-методичного комплексу, який би відповідав сучасному стану розвитку чисельних методів та технології програмування.

Традиційні курси чисельних методів, що викладаються з використанням процедурної методології, на алгоритмічному рівні оперують з математичними абстракціями високого рівня – векторами, матрицями, поліномами і т.ін., та при переході до програмної реалізації методу основний час витрачається не на сам метод, а на реалізацію операцій векторної, матричної та поліноміальної алгебр. Це приводить до 1) "розбухання" тексту програми; 2) чіткий алгоритм чисельного методу в програмі втрачає свою ясність, прозорість; 3) елементарні операції над такими об'єктами ведуть до великої кількості вкладених циклів тощо. Основною причиною цих вад є *відсутність у процедурній методології засобів для адекватного подання абстракцій високого рівня* у вигляді нових, побудованих користувачем типів даних, операції над якими виконуються так само, як і над вбудованими типами даних.

Наш досвід свідчить, що впровадження об'єктного підходу в практику вивчення чисельних методів у вищій школі дозволяє позбутися зазначених недоліків і має великий потенціал для формування активної пізнавальної діяльності студентів. Цьому сприяють можливість роботи з математичними абстракці-

ями різного рівня, їх ієрархічність та модульність, паралелізм та багатоваріантність програмного проектування, які забезпечує об'єктний підхід.

Застосування об'єктного підходу до викладання чисельних методів приводить до максимального зближення програмної реалізації методу та його алгоритмічного запису завдяки наявності механізму створення нових типів даних та перевизначення для них стандартних операцій. Це дозволяє, наприклад, операцію множення двох матриць відповідної розмірності записати так само, як і в алгоритмі: A*B, а не за допомогою трьох вкладених циклів, як це доводиться робити у процедурній методології. Таке спрощення тексту програм інтенсифікує вивчення курсу, дозволяючи викласти матеріал стисліше, обмежуючись лише розглядом алгоритму методу без розгляду його програмної реалізації. Зменшення трудозатрат на програмування методу супроводжується ще одним ефектом: у активну навчально-пізнавальну діяльність включаються навіть ті студенти, які мають недостатню базову підготовку з програмування.

На наш погляд, саме в курсі чисельних методів найбільш повно реалізуються потужні можливості об'єктного підходу, які в інших курсах поки що не знаходять відповідного відображення. Це вимагає подальшої розробки цієї проблеми, не обмежуючись лише об'єктно-орієнтованим програмуванням, а досліджуючи можливості самого об'єктного підходу до викладання різних дисциплін.

В основу об'єктного підходу покладено методи декомпозиції, виділення абстракцій та створення ієрархій. Об'єктний підхід утворює концептуальний базис об'єктно-орієнтованої методології. Її втіленням є *об'єктно-орієнтоване програмування* – методологія програмування, заснована на поданні програми у вигляді сукупності об'єктів, кожен з яких є реалізацією деякого класу, а класи утворюють ієрархію за принципом наслідування.

На нашу думку, з позицій підтримуваних абстракцій мови програмування можна класифікувати так: *математичні, алгоритмічні, орієнтовані на дані та об'єктно-орієнтовані*. Автор об'єктно-орієнтованої мови програмування C++ Б. Страуструп у [9] зазначає: "… якщо термін "об'єктно-орієнтована мова" взагалі щось означає, то він повинен означати мову, яка має засоби гарної підтримки об'єктно-орієнтованого стилю програмування… Забезпечення такого стилю у першу чергу означає, що у мові зручно використовувати такий стиль. Якщо написання програм у стилі ООП вимагає спеціальних зусиль чи воно неможливе зовсім, то ця мова не відповідає вимогам ООП". Теоретично можливою є імітація об'єктно-орієнтованого програмування навіть мовою ассеблера (як це показано у [3]), проте це надзвичайно важко здійснити практично. Загальноприйнятим є підхід, за яким мова програмування вважається об'єктно-орієнтованою, якщо виконуються такі умови:

- підтримуються об'єкти, тобто абстракції даних, що мають інтерфейс у вигляді іменованих операцій, та власні дані з обмеженням доступу до них;
- об'єкти відносяться до відповідних типів (класів);
- типи (класи) можуть наслідувати атрибути супертипів (суперкласів).

Для уникнення термінологічної неоднозначності дамо такі означення:

1. *Об'єктними* називають мови, які підтримують абстракцію даних та класи.

2. *Об'єктно-орієнтованими* називають ті об'єктні мови, які підтримують наслідування та поліморфізм.

Кожний стиль програмування має свою концептуальну основу; для об'єктно-орієнтованого стилю цією основою є об'єктний підхід, принципи якого були сформульовані Г. Бучем у [1]. Застосування цих принципів дозволило нам виконати порівняльний аналіз основних мов програмування, який узагальнено у таблиці:

	Smalltalk	**Object Pascal**	**C++**	**Common Lisp Object System (CLOS)**	**Ada**	**Eiffel**
Абстракції						
Змінні об'єкту	Так	Так	Так	Так	Так	Так
Методи об'єкту	Так	Так	Так	Так	Так	Так
Змінні класу	Так	Ні	Так	Так	Ні	Ні
Методи класу	Так	Ні	Так	Так	Ні	Ні
Інкапсуляція						
Змінних	Приватні	Загальнодоступні	Загальнодоступні, захищені, приватні	Читання, запис, доступ	Загальнодоступні, приватні	Приватні
Методів	Загальнодоступні	Загальнодоступні	Загальнодоступні, захищені, приватні	Загальнодоступні	Загальнодоступні, приватні	Загальнодоступні, приватні
Модульність						
Різновиди модулів	Ні	Модуль (інтерфейс – реалізація)	Файл (заголовок – тіло)	Пакет (монолітний)	Пакет (специфікація – тіло)	Блок (інтерфейс – реалізація)
Ієрархії						
Наслідування	Одиночне	Одиночне	Множинне	Множинне	Ні (входить у Ada9x)	Множинне
Шаблони	Ні	Ні	Так	Ні	Так	Так
Метакласи	Так	Ні	Ні	Так	Ні	Ні
Типізація						
Сильна типізація	Ні	Так	Так	Можлива	Так	Так

	Smalltalk	**Object Pascal**	**C++**	**Common Lisp Object System (CLOS)**	**Ada**	**Eiffel**
Полі-морфізм	Так (одиночний)	Так (одиночний)	Так (одиночний)	Так (множинний)	Ні (входить у Ada9x)	Так
Паралелізм						
Багатозадачність	Непряма (за допомогою класів)	Ні	Непряма (за допомогою класів)	Так	Так	Ні
Стійкість						
Довгоживучі об'єкти	Ні	Ні	Ні	Ні	Ні	Ні
Тип мови	Чисто об'єктно-орієнтована	Спрощена об'єктно-орієнтована	Гібридна об'єктно-орієнтована	Гібридна об'єктно-орієнтована	Об'єктна (об'єктно-орієнтована у Ada9x)	Об'єктно-орієнтована

Проведений аналіз дозволяє зробити висновок, що найбільш придатною мовою для навчання програмування у об'єктній методології є мова C++. Ця мова є гібридною і це дозволяє організувати неперервне навчання студентів процедурній та об'єктній методології в межах однієї мови. Спочатку вивчається підмножина C++, що містить основні конструкції мови C, а далі на цьому базисі надбудовуються компоненти, які надають цій мові об'єктно-орієнтованих властивостей. Чітка структуризація системи захисту разом з наявністю трьох рівнів доступу до програмного коду та даних дозволяє будувати логічно завершені системи класів та їх модулів.

Об'єктно-орієнтоване проектування – це методологія проектування, що поєднує в собі процес об'єктної декомпозиції та прийоми подання логічної та фізичної, а також статичної та динамічної моделей проектованої системи.

Саме об'єктно-орієнтована декомпозиція відрізняє об'єктно-орієнтоване проектування від структурного: у першому випадку логічна структура системи відображається абстракціями у вигляді класів і об'єктів, у другому – алгоритмами.

Об'єктно-орієнтований аналіз – це методологія, при якій вимоги до системи сприймаються з точки зору класів та об'єктів, виявлених у предметній області. Об'єктно-орієнтований аналіз спрямований на створення моделей реальної дійсності на основі об'єктно-орієнтованого світогляду.

Між всіми цими визначеннями існує тісний взаємозв'язок: за результатами об'єктно-орієнтованого аналізу створюються моделі, на яких грунтується об'єктно-орієнтоване проектування; у свою чергу, об'єктно-орієнтоване проек-

тування створює фундамент для фінальної реалізації системи з використанням методології ООП.

А.П. Єршов у своїй роботі “Про об’єктно-орієнтовану взаємодію з ЕОМ” [2] надає об’єктно-орієнтованому програмуванню більш широкого змісту, ніж програмуванню з використанням об’єктно-орієнтованих мов. У якості одного з прикладів об’єктно-орієнтованої взаємодії програмуючого користувача з ЕОМ А.П. Єршов посилається на Е-практикум як на реальну систему автоматизованого конструювання програм. Особливо він наголошує на тезі про *перспективність та універсальність об’єктно-орієнтованої взаємодії*, основними компонентами якої є об’єктно-орієнтоване проектування та аналіз. Як розвиток цієї думки А.С. Лесневський у [5] пропонує в термінах об’єктного підходу розв’язання задачі моделювання поведінки виконавця-робота, який блукає лабіринтом, зображеним на екрані. При цьому мовою програмування було обрано русифікований Смолток, а середовищем – об’єктний варіант Е-практикума.

А. Співаковский наводить приклад ППЗ на основі об’єктно-орієнтованого підходу. Він звертає особливу увагу на навчальний ефект таких ППЗ, які “надають користувачеві свободу, обмежену лише рамками предметної області” [7]. А.Б. Кузнєцов у [4] виділяє аспекти, згідно яких навчання об’єктно-орієнтованого програмування у класах з поглибленим вивченням інформатики є бажаним: *концептуальність об’єктно-орієнтованого підходу*, *наявність соціального замовлення на вивчення в школі об’єктно-орієнтованого програмування* та *підтримка інтересу учнів до програмування*.

Таким чином, об’єктно-орієнтований підхід має великий навчальний потенціал, що й зумовило його поступове впровадження в практику навчання як середньої, так і вищої школи.

Досвід викладання автором курсу чисельних методів в об'єктній методології, узагальнений у навчальному посібнику [6], показав, що застосування об’єктно-орієнтованого підходу до розробки спеціалізованих бібліотек математичних об'єктів істотно підвищує рівень самостійної пізнавальної активності студентів як на етапі аналізу і проектування бібліотеки, так і при програмній реалізації методу. Дійсно, оскільки об’єктно-орієнтоване проектування сприяє багатоваріантності способів побудови бібліотеки, то й конструювання запропонованої бібліотеки від самого початку студенти мимоволі здійснювали різними шляхами. Це приводило до розмаїтості індивідуальних класів і робило неможливий плагіат: адже для розуміння сутності побудованих кимось класів необхідно мислено пройти весь шлях аналізу і проектування.

При створенні власних бібліотек математичних об'єктів студенти в багатьох випадках творчо підходили до поставленій задачі. Вони не обмежувалися векторними, матричними і поліноміальними класами, побудова яких була обов'язкової, а створювали розгалужені системи класів як за ієрархією спадкування, так і за ієрархією спорідненості, іноді далеко відходячи від поставленої задачі.

Яскравим прикладом такої діяльності є робота, виконана однією зі студентських груп. Основна мета їхньої діяльності складалася в розробці комп'ютерних інтерпретацій алгебраїчних структур і нових типів даних, що розширюють

можливості мови C++ щодо дослідження теоретико-числових проблем і проведення розрахунків високої точності. Виконавши об’єктно-орієнтований аналіз, студенти дійшли таких висновків:

1) неможливість виконання засобами машинної арифметики операцій над числами необмеженої довжини (точності) веде до необхідності створення відповідних бібліотек для роботи з такими об'єктами;

2) наявність загальних підходів до конструктивної побудови алгебраїчних систем веде до надбудови однієї алгебри над інший так само, як і наслідування властивостей одного класу від іншого.

Така природна спільність класів мови C++ і алгебраїчних структур дала можливість побудувати комп'ютерні інтерпретації основних числових систем у виді розгалуженої ієрархії математичних класів, що дозволило використовувати в програмі нові типи даних – “довгих” цілих і раціональних чисел так само, як і звичайні машинні типи даних обмеженої довжини, змішуючи їх. Повернувшись до основної задачі, студенти застосували створені числові об'єкти при побудові параметризованого класу арифметичних векторів, компонентами якого можуть бути числові об'єкти будь-якої природи, зокрема – побудовані класи необмежених цілих і раціональних чисел. Це дозволило змоделювати такі числові об'єкти, як кільце комплексних чисел з раціональними компонентами, тіло кватерніонів і т.п., знову вийшовши за границі поставленої задачі.

Література:

1. Буч Г. Объектно-ориентированное проектирование с примерами применения. – М.: И.В.К., К.: Диалектика, 1992. – 519 с.
2. Ершов А.П. Об объектно-ориентированном взаимодействии с ЭВМ // Микропроцессорные средства и системы. – 1985. – № 3. – С. 2.
3. Использование Turbo Assembler при разработке программ. – К.: Диалектика, 1993. – 288 с.
4. Кузнецов А.Б. Программа курса “Основы объектно-ориентированного программирования” // Информатика и образование. – 1998. – № 7. – С. 17-24.
5. Лесневский А.С. Практикум по объектно-ориентированному проектированию и программированию // Информатика и образование. – 1998. – № 5. – С. 114-121.
6. Полищук А.П., Семериков С.А. Методы вычислений в классах языка C++: Учебное пособие. – Кривой Рог: Издательский отдел КГПИ, 1999. – 350 с.
7. Спиваковский А. Педагогические программные средства: объектно-ориентированный подход // Информатика и образование. – 1990. – № 2. – С. 71-73.
8. Хемминг Р.В. Численные методы для научных работников и инженеров. – М.: Наука, 1968. – 400 с.
9. Stroustrup B. What is object-oriented programming? – IEEE Software. – 1988. – Vol. 5. – P. 10-20.