

С.О. Семеріков

Криворізький державний
педагогічний університет

Чисельні методи: об'єктний підхід

1. Об'єктний підхід як метод дослідження складних систем: логіко-психологічний аспект

Об'єктно-орієнтоване програмування (ООП), що отримало широке розповсюдження як потужна програмна технологія, є у наш час вагомою альтернативою традиційним процедурним методам програмування. Популярність ООП у чималій мірі визначається концептуальною цілісністю та більш сильною формою структуризації програмного забезпечення (ПЗ), що створюється на його основі. Використання ООП прискорює процес розробки програм, даючи при цьому можливість гнучкої та природної модифікації існуючого ПЗ. Найбільш рельєфно можливості ООП проявляються при створенні досить складних програмних продуктів, до яких, зокрема, відносяться проблемно-орієнтовані бібліотеки. Розглянемо основні передумови впровадження об'єктного підходу.

Основне утруднення, що постає перед розробником складного програмного продукту, полягає в неможливості одночасно утримувати в пам'яті усі необхідні деталі. Психологи стверджують, що максимальна кількість одиниць інформації, з якою здатен одночасно оперувати людський мозок, не перевищує 7 ± 2 (так званий "гаманець Міллера"). Ще одним обмежувачим фактором тут виступає швидкість обробки мозком нової інформації: йому потрібно близько 5 с на кожен нову подію. Як бачимо, природна здатність людського мозку до роботи із складними системами є низькою.

Проте, як зазначає Б. Страуструп [4], ще з давніх давен людству відомий простий та ефективний спосіб управління складними системами: "Розділяй та володарюй". Тому при

проектуванні складної системи (програми) необхідно складати її з окремих невеликих підсистем (підпрограм) – у цьому випадку ми не виходимо за межі можливостей людини: при розробці будь-якого рівня системи необхідно одночасно утримувати в пам'яті інформацію лише про деякі її частини. Такий підхід забезпечує психологічне підґрунтя для процедурного програмування, визначаючи головну вимогу до написання підпрограми: “усі дії, що виконуються в підпрограмі, повинні усвідомлюватися *одночасно*”, і якщо ця вимога не виконується, підпрограму слід розбити на дрібніші блоки. До того ж у шкільному віці “гаманець Міллера” має менші розміри, тому при вивченні основ алгоритмізації поняття допоміжного алгоритму (процедури) та алгоритмічної декомпозиції, на наш погляд, повинні вводитися одними з перших, а самі процедури не повинні містити більше 4–5 усвідомлюваних дій.

Проте число подій, що одночасно може опрацювати людина, *не залежить від обсягу інформації*, що міститься у кожній події, і це дає людині надзвичайно ефективний механізм обробки складних повідомлень – *абстрагування*. Не маючи можливості відтворити у всіх деталях складний об'єкт, ми ігноруємо несуттєві для нас деталі і, таким чином, маємо справу з узагальненою, ідеалізованою моделлю об'єкта. І хоч при цьому ми також обмежені кількістю інформації, яку можемо сприйняти у кожний окремий момент, все одно, використовуючи абстрактні поняття, ми отримуємо можливість працювати із складними системами, а, отже, і створювати складні програмні продукти.

Складні системи можна досліджувати, концентруючи основну увагу або на об'єктах, що фігурують у системі, або на процесах, що протікають в ній. Проте доцільніше розглядати систему як впорядковану сукупність об'єктів, які в процесі взаємодії один з одним забезпечують функціонування системи як єдиного цілого. Об'єкти, що складають систему, можуть утворювати *ієрархії*. При такому підході основним способом дослідження складної системи є *об'єктна декомпозиція*.

Таким чином, з'являється можливість розширити границі когнитивних можливостей людини, використовуючи методи декомпозиції, виділення абстракцій та створення ієрархій. Саме ці методи покладено в основу *об'єктного підходу*, який утворює концептуальний базис *об'єктно-орієнтованої методології*.

Об'єктний підхід відомий ще з давніх часів. Так, давнім грекам належить ідея про те, що світ можна розглядати як у термінах об'єктів, так і подій. А у XVII ст. Декарт підкреслював, що люди зазвичай мають об'єктно-орієнтований погляд на світ. У 60–70-х р.р. XX ст. ця думка була розвинена в одній з течій когнитивної філософії – об'єктивістській епістеміології (Е. Ранд), а на початку 80-х р.р. М. Мінські запропонував модель людського мислення, у якій розум людини розглядається як спільнота агентів, що по-різному мислять. На його думку, лише спільні дії таких агентів приводять людину до осмисленої поведінки.

Відомий американський спеціаліст в галузі ООП Г. Буч у [1] виділяє такі принципи об'єктного підходу:

Абстрагування – виділення таких суттєвих характеристик об'єкту, які відрізняють його від усіх інших об'єктів і, таким чином, чітко визначають особливості даного об'єкту з точки зору його подальшого розгляду.

Обмеження доступу – процес захисту окремих елементів об'єкту, що не зачіпає суттєвих характеристик об'єкту як цілого.

Модульність – властивість системи, пов'язана з можливістю її декомпозиції на ряд тісно пов'язаних модулів.

Ієрархія – рангована чи впорядкована система абстракцій.

Типізація – обмеження, що пред'являється до класу об'єктів, перешкоджаючи взаємозаміні різних класів та звужуючи можливість такої взаємозаміни.

Паралелізм – властивість, що відрізняє активні об'єкти від неактивних.

Стійкість – властивість об'єктів існувати у часі та/або у просторі.

Реалізацією об'єктно-орієнтованої методології дослідження складних систем є *об'єктно-орієнтоване програмування* – методологія програмування, заснована на представленні програми у вигляді сукупності об'єктів, кожен з яких є реалізацією деякого класу, а класи утворюють ієрархію за принципами наслідуваності.

Таким чином, об'єктно-орієнтоване програмування є найбільш природною методологією програмування, яка, враховуючи особливості психічних процесів, дає можливість створювати чітко структуровані та осяжні складні програмні продукти.

2. Застосування об'єктного підходу до розробки математичного ПЗ: «за» без «проти»

З початку 90-х р.р. ведуться інтенсивні дослідження у галузі створення об'єктно-орієнтованих операційних систем, баз даних, графічних інтерфейсів. Значно менше уваги приділяється питанням застосування об'єктних технологій до математичного ПЗ, орієнтованого на розв'язання задач обчислювального характеру. До цих пір найбільш широкоживаною мовою для розв'язання таких задач вважається Фортран. Різноманітність математичних та прикладних бібліотек, створених цією мовою, а також безсумнівні традиції є суттєвими факторами, що стримують впровадження об'єктних технологій у цю галузь. Розглянемо найбільш важливі особливості процедурної та об'єктної реалізації математичного ПЗ, виходячи з традиційної технології використання універсальних математичних бібліотек.

Як зазначає В.А. Семенов [3], першою перепорою до інтеграції ПЗ є необхідність уніфікації всіх тих внутрішніх форматів даних програми, які будуть використовуватись імпортованими бібліотечними модулями. Це не завжди можливо, оскільки часто викликає необхідність розширити методичний

репертуар однієї універсальної бібліотеки спеціалізованими засобами з інших бібліотек. Єдиною можливістю формального об'єднання модулів у цьому випадку є одночасна підтримка декількох еквівалентних форматів даних на основі їх попереднього уважного вивчення. Зауважимо, що навіть у випадку вдало специфікованих форматів різних бібліотек розробник все одно підтримує усі внутрішні дані, маючи до них безпосередній доступ. Такий підхід часто стає джерелом помилок і, безумовно, не може вважатися задовільним для великих програмних проєктів.

Другим не менш суттєвим недоліком процедурного підходу є низька модифікованість програмних модулів. Навіть незначна зміна однієї з алгоритмічних гілок у обчислювальному модулі, як правило, вимагає детального ознайомлення з організацією усього обчислювального процесу та зі структурою усієї програми і часто пов'язана із значними зусиллями по перепрограмуванню. З урахуванням вкрай низької наочності та виразності процедурних мовних засобів така техніка є доступною лише висококваліфікованим спеціалістам як у галізі обчислювальних методів, так і програмування.

Типовими прикладами подібних модифікацій можуть бути, зокрема, або зміна стратегії вибору головного елемента в методах факторизації, коли існує додаткова апріорна інформація про матричну обумовленість чи розрідженість, або посилення умов зупинки ітераційного процесу для більш надійного контролю збіжності у нелінійних алгебраїчних задачах, або зміна евристичного сценарію вибору крока при чисельному інтегруванні диференціальних рівнянь деякого виду і т.п.

Суттєвим є те, що орієнтація лише на процедурну завершеність не сприяє внутрішній структуризації програм та надзвичайно утруднює їх модифікацію та використання. Застосування ж концепції ООП для розробки математичного забезпечення представляється надзвичайно привабливим. Система класів (або об'єктна бібліотека), що є фундаментальною формою об'єктно-орієнтованого ПЗ, виявляється потужним та

виразним засобом реалізації математичних бібліотек та прикладних обчислювальних систем.

Розглянемо головні можливості, що їх надає об'єктна технологія при реалізації обчислювального математичного забезпечення.

Перш за все, *інкапсуляція* методів (в даному випадку – чисельних методів) у класах математичних об'єктів є більш сильною формою структуризації обчислювальних модулів, яка підвищує концептуальну наочність застосовуваних чисельних методів та регламентує коректну дисципліну роботи з даними об'єктів без порушення їх цілісності. Розробник прикладного ПЗ виступає користувачем бібліотечних класів і замість підтримки внутрішніх даних може зосередити зусилля на предметному аспекті розроблюваних програм. Приймаючи до уваги складність динамічно розміщуваних даних математичних об'єктів, зокрема, розріджених матричних об'єктів, зазначена перевага є більш, ніж помітною.

Механізм *захисту даних*, що забезпечує об'єктна технологія, також сприяє підвищенню надійності розроблюваних програм.

Нарешті, *наслідування* математичних об'єктів та *поліморфізм* чисельних методів, інкапсульованих ними, надає важливу для багатьох чисельних застосувань можливість гнучкої модернізації і розвитку математичного забезпечення, як з урахуванням проблемної орієнтації, так і у відповідності до детальної класифікації об'єктів за їх математичними та обчислювальними ознаками і властивостями. Реалізація чисельних методів у вигляді наслідуваних методів споріднених математичних класів дозволяє *природним способом* виразити міру спільності математичних понять та ступінь універсальності застосування тих чи інших чисельних підходів у кожному конкретному випадку.

Таким чином, реалізація математичного забезпечення у вигляді бібліотеки чи системи класів має суттєві переваги над традиційною процедурно-орієнтованою формою ППЗ.

3. Активізація пізнавальної діяльності при вивченні чисельних методів у об'єктній методології: теорія, методика, практика

Одне з основних методичних утруднень, що виникає при викладанні курсу чисельних методів, полягає у необхідності одночасного засвоєння обчислювальних алгоритмів та особливостей їх програмної реалізації в умовах дефіциту навчального часу.

Традиційно, при вивченні обчислювального алгоритму ми оперуємо з абстракціями високого рівня – поліномами, матрицями, векторами тощо. Та при програмній реалізації у процедурній методології навіть прості операції з такими об'єктами породжують громіздкі конструкції з безліччю вкладених циклів, що віддаляє програму від алгоритма, а реалізацію – від метода, і це є першим аспектом проблеми. Другий її аспект – це повторюваність одних й тих самих процедур у різних розділах чисельних методів. Обидва аспекти в поєднанні із слабко вираженою практичною спрямованістю традиційних курсів обчислювальної математики призводять до того, що інколи студент навіть не здогадується, навіщо він вивчає той чи інший метод.

Поширеним підходом при викладанні чисельних методів є використання спеціалізованих математичних бібліотек, проте існуючі зараз універсальні математичні бібліотеки є в основному результатом тривалої еволюції процедурного програмування і мають досить обмежені можливості для подальшого розвитку. Об'єктний підхід забезпечує більш радикальні засоби для бажаної інтеграції і модифікації ПЗ.

Природна спільність принципів ООП і методологічних основ обчислювальної математики була досліджена наприкінці 90-х р.р. у роботах В.А. Семенова, Р. Позо, Й. Арндта, Б. Хайбле та інших, проте застосування ООП до викладання відповідних дисциплін і сьогодні залишається майже недослідженою проблемою. Оскільки класи математичної бібліоте-

ки реалізують найзагальніші проблемно-інваріантні поняття, бібліотека класів може розглядатися як базове інструментальне середовище для розробки курсу чисельних методів. При цьому інтерфейс бібліотечних класів утворює своєрідну мову, максимально наближену до природної математичної, використання якої зближує етапи алгоритмічного і програмного проектування і істотно полегшує програмування обчислювальних задач.

Розглянемо базові аспекти методики викладання курсу чисельних методів у об'єктній методології [2], розроблену та апробовану у Криворізькому педагогічному університеті.

На першому етапі будується базис у вигляді математичних класів векторів, поліномів та матриць, які інкапсулюють у собі операції над елементами відповідних множин та типові процедури обробки даних, що базуються на цих операціях (від методу найменших квадратів та обчислення поліноміальних нулів до розв'язання задач лінійного та динамічного програмування). Після завершення цієї роботи студенти одержують можливість записувати у своїх програмах операції над відповідними типами у природній математичній нотації.

На другому етапі довільній групі чисельних методів ставиться у відповідність конкретна прикладна задача, яка вимагає їх використання. Так, при вивченні методів розв'язання диференціальних рівнянь ставиться задача комп'ютерного моделювання поведінки лінійної динамічної системи з розрахунком вільних та вимушених рухів під дією стандартизованих збурень типу імпульсного та ступінчатого впливів, гармонічних функцій та довільних функцій часу. Задача розв'язується як за допомогою програмної реалізації аналітичного операторного методу, так і за допомогою кінцево-різницевої схем Адамса-Башфорта-Моултона з можливістю порівняння результатів. При цьому інтенсивна робота з комплексними, матричними, векторними та поліноміальними об'єктами не створює жодних технічних утруднень за рахунок попередньо проведеної роботи. Це дозволяє досягти макси-

мального наближення тексту програмної реалізації чисельного методу до його алгоритмічного опису.

Нами розроблено такі класи для мов C++ та Object Pascal. Проте слід відмітити переваги C++ як мови, що підтримує механізм перевизначення операцій та функцій. Це дозволяє, наприклад, операцію додавання двох матриць однакового порядку записати у вигляді $matrix\ c=a+b$. Інший приклад: якщо визначити у матричному класі унарну операцію транспонування \sim , бінарну операцію множення $*$ та унарну операцію знаходження оберненої матриці $\&$, то при заданій матриці вхідних впливів x та векторі спостережень у розв'язок задачі знаходження вектора a коефіцієнтів моделі за МНК у тексті програми мовою C++ запишеться так:

$$matrix\ a=(\&(\sim x*x))*(\sim x*y);$$

у той час як у Паскаль-реалізації він матиме такий вигляд:

$$a=Mul(PowMinusOne(Mul(Transp(x),x)),Mul(Transp(x),y));$$

Підсумовуючи, зазначимо, що:

1. Наявність готової бібліотеки математичних об'єктів суттєво прискорює процес програмної реалізації метода, скорочуючи не лише витрати часу на написання програми, а й її обсяг, робить її більш «прозорою» за рахунок підвищення рівня абстракції до операцій над новими типами даних. Параметризація програм дозволяє породжувати з шаблонів типів спеціалізовані реалізації простою підстановкою типу.

2. Застосування розробленої бібліотеки дозволило розширити традиційний курс чисельних методів розділами, програмна реалізація яких засобами процедурного програмування звичайно викликає утруднення.

3. Апробація курсу «Чисельні методи в об'єктній методології» протягом останніх років свідчить про підвищення якості засвоєння навчального матеріалу за рахунок переключення уваги студентів з деталей програмної реалізації на сам метод завдяки наближенню програмного запису алгоритма до природного математичного.

Литература:

1. Буч Г. Объектно-ориентированное проектирование с примерами применения. – М.: Конкорд, 1992.
2. Полищук А.П., Семериков С.А. Методы вычислений в классах языка С++: Учебное пособие. – Кривой Рог: Издательский отдел КГПИ, 1999. – 350 с.
3. Семенов В.А. Объектно-ориентированная методология эволюционной разработки математического обеспечения. Диссертация ... доктора физ.-мат. наук. – М.: ИСП РАН, 1998.
4. Страуструп Б. Язык программирования С++. Ч. 1, 2. – К.: ДиаСофт, 1993.