

Юссеф Т.Х.

студентка,

*Одеський державний університет
імені І.І. Мечникова*

ВИКОРИСТАННЯ ООП ПРИ РОЗВ'ЯЗУВАННІ ПРИКЛАДНИХ ЗАДАЧ ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ

Аналіз фізичних процесів та їх математичних моделей нерозривно пов'язаний з такими поняттями, як функція, графік функції, таблиця функції.

Теоретичні дослідження, як правило, оперують із функціями, для яких відомо (або задається) їх аналітичний вигляд. Наприклад, закон гармонійного коливання матеріальної точки є наступною функцією часу: $x(t) = a \times \sin(\omega t + \alpha)$.

Інший типовий випадок – теоретична залежність в'язкості моторного мастила μ від температури його розігріву θ , яка описується теоретичною формулою (функцією) $\mu(\theta) = \mu_0 \times \exp\left(-\delta_0 \times \frac{\theta}{1+\theta}\right)$.

На підставі цих аналітичних залежностей здійснюється подальший математичний аналіз фізичних моделей, формуються загальні теоретичні висновки та технологічні рекомендації щодо їх застосування.

Принципово інша ситуація виникає, коли фізичне явище (технологічний процес) або його математична модель досліджуються через натурний (або комп'ютерний) експеримент.

У цьому випадку, у розпорядженні дослідника є лише числові дані (результати експерименту), що описують явище. Ці дані він мусить якось обробити заради досягнення кінцевої мети дослідження.

Як правило, такі числові дані зберігаються у вигляді (формі) таблиць, які дискретним чином визначають ті або інші фізичні залежності.

Методи наближених обчислень повинні забезпечити нас можливістю «обробки» функцій в обох зазначених формах: у вигляді аналітичної залежності $f = f(x)$, та у формі дискретно заданої функції, яку записано у вигляді упорядкованої таблиці числових даних.

Для розв'язування зазначеної проблеми нами пропонується використання широких можливостей, які має об'єктно-орієнтоване програмування [1; 2], та які нам надає сучасне середовище розробки Microsoft Visual Studio та популярна мова програмування C#.

Традиційно програмні реалізації класичних методів обчислювальної математики об'єднуються у динамічні бібліотеки *.dll, в яких однорідні чисельні методи та похідні типи даних формують окремі класи або структури.

Саме такої стратегії ми дотримувались у розробці нашої бібліотеки.

Одразу зазначимо, що основною метою даної роботи було продемонструвати можливості та основні принципи ООП, такі як: інкапсуляція, поліморфізм та наслідування, які роблять процес розробки прикладного програмного забезпечення більш гнучким, зрозумілим та комфортним.

Важливо те, що студент, який буде опановувати даний підхід, не тільки засвоїть алгоритми обчислювальної математики, а й удосконалив свої навички у сучасному програмуванні. Він матиме певну кількість прикладів програмування тих або інших проблем, ознайомиться із способами роботи з різноманітними типами даних, включаючи розробку власних типів даних.

Наша реалізація бібліотеки чисельних методів ґрунтується на тому, що основним елементом, який треба математично обробляти є таблиця функції.

Вище вже зазначалося, що є два принципово різні за своєю природою шляхи «генерації» таблиці функції: із аналітичної формули $f = f(x)$, або із натурального (чисельного) експерименту у вигляді певного набору числових даних.

Реалізація загального підходу до даної проблеми полягає в тому, що є спроектований абстрактний клас `MyTable`, в якому були задекларовані загальні члени, які властиві обома типам таблиць та мають спільні методи обробки.

На основі цього класу створені два класи-спадкоємці: `MyTableOfData` та `MyTableOfFunction`, які описують зазначені типи таблиць функції.

Кожен з цих класів має свій власний конструктор (або конструктори), який розбудовує первинну таблицю функції за певними правилами, що враховують специфіку даного типу таблиці.

Окремо в цих класах додаються методи, які скасовують віртуальні методи класу `MyTable` та визначають їх власні реалізації.

Також додаються певні методи, які реалізують алгоритми обчислювальної математики із врахуванням особливостей розбудови структури даних цих таблиць функцій.

Слід окремо зазначити, що об'єкт класу `MyTableOfData` набуває свого змісту із файлів даних, які за правилами нашої бібліотеки, повинні бути попередньо створені у тестовому форматі, або записані у бінарній формі, як результати певних обчислень (чисельного експерименту).

Ці файли інколи можуть мати досить великий обсяг числової інформації.

Текстові форматні файли (невеликі за обсягом) дозволяють досліднику швидко переглядати результати експерименту та робити попередні висновки на підставі їх безпосереднього візуального аналізу.

Однак ці файли не зручні для їх подальшого використання в рамках інших додатків (програм), оскільки потребують додаткових зусиль для організації алгоритмів зчитування числової інформації із подальшою конвертацією у набори даних числового типу, над якими слід виконувати певні математичні дії.

Файли бінарного типу зберігають інформацію у «внутрішній» формі, що відповідає архітектурі даного конкретного процесора (комп'ютера). Тому без будь-яких додаткових зусиль дані з цих файлів можна завантажувати у оперативну пам'ять комп'ютера.

Крім цього, при одному й тому ж об'ємі корисної математичної інформації бінарні файли мають розміри в кілька разів менші, ніж файли текстові.

Додатково до класу `MyTable` розроблено окремі класи, які узагальнюють такі математичні поняття, як вузол таблиці функції, інтервал з області визначення функції, що містить нуль цієї функції, та інші.

Використання таких допоміжних класів значно полегшує програмну реалізацію багатьох математичних алгоритмів, наприклад, пошук вузла таблиці із найбільшим, або найменшим значенням функції, чи ітераційний алгоритм уточнення нулів функції.

Окремо розроблялася ціла низка програмних реалізацій методів обчислювальної математики [3; 4], серед яких слід виділити: методи уточнення коренів нелінійних рівнянь; методи інтерполювання функцій; методи обчислення визначених інтегралів; методи інтегрування задачі Коші для загальних диференціальних рівнянь; методи регресійного аналізу даних.

Всі розроблені нами програмні компоненти перевірялися на окремих задачах та тестувалися для обох типів таблиць функції.

Для користування цими програмними компонентами створене відповідне програмно-методичне забезпечення для студентів фізико-математичних спеціальностей ОНУ імені І.І. Мечникова.

До цього забезпечення (окрім стислого конспекту теоретичного матеріалу з методів наближених обчислень) входить низка лабораторних робіт з тестовими задачами (які демонструють очікувані результати) та контрольних завдань для індивідуального виконання.

Список використаних джерел

1. Карли Ватсон С# : Издательство «Лори», 2006. – 860 с.
2. Эндрю Троелсен Язык программирования С# 2005 и платформа .NET 2.0, 3-е издание.: Пер. с англ. – М.: ООО «И.Д.Вильямс», 2007. – 1168 с.
3. Коссак О. Методи наближених обчислень: навч. посібник / Коссак О., Тумашова О., Коссак О. – Львів: БаК, 2003. – 168 с.
4. Шахно С.М. Практикум з чисельних методів: навч. посібник / Шахно С.М., Дудикевич А.Т., Левицька С.М. – Львів: ЛНУ імені Івана Франка, 2013. – 432 с.