

УДК 004.4'242

## **ДИНАМІЧНА ГЕНЕРАЦІЯ КОДУ НА МОВІ ПРОГРАМУВАННЯ C# НА ПРИКЛАДІ РІШЕННЯ НЕЛІНІЙНОГО РІВНЯННЯ МЕТОДОМ ХОРД**

Кравченко В.А., 11 МБКН,

Гавриленко Є.А., к.т.н., доц.

*Таврійський державний агротехнологічний університет*

тел.: (0619) 42-68-62.

***Анотація*** - робота присвячена дослідженню можливостей методу динамічної генерації та запуску програмного коду на мові програмування C# на прикладі рішення нелінійного рівняння методом хорд під час виконання додатка.

***Ключові слова*** – CodeDOM, C#, програмний код, динамічна генерація, нелінійні рівняння, метод хорд.

*Постановка проблеми.* Тлумачення терміну «штучний інтелект» на сьогоднішній час обумовлює декілька варіантів, більшість з яких істотно різняться, навіть у представників суміжних професій. Наприклад, практичні програмісти, мають на увазі під цим поняттям програму, здатну переписувати (модифікувати) саму себе. Тобто, при необхідності і в залежності від вимог користувача або розв'язуваної задачі, додаток буде здатний змінювати окремі частини власного програмного коду, підлаштовуючись під нові вимоги.

*Аналіз останніх досліджень.* На теперешній час загально відомі інструменти, за допомогою яких можна реалізувати таку систему. Один з найпотужніших – це динамічна генерація коду, яку, поступово, починають застосовувати в розробці окремих програмних продуктів. Звичайно, можливості до такого підходу досить широкі, проте перш ніж приступити до створення масштабної системи, здатної модифікувати саму себе і вчитися таким чином, необхідно детально

вивчити особливості методів автоматичної генерації програмного коду на більш простих випадках.

*Формулювання цілей статті.* Ознайомлення з базовими можливостями методу динамічної генерації програмного коду на прикладі вирішення задачі з визначення коренів нелінійних рівнянь за допомогою об'єктно-орієнтованої мови програмування C#..

*Основна частина.* Одним з найбільш підходящих інструментів для вирішення подібних завдань є об'єктно-орієнтована мова програмування C#. Вона була створена Microsoft як мова розробки додатків для платформи Microsoft .NET Framework [1, с 14]. Однією з особливостей C# є те, що додатки .NET не інтерпретуються, а компілюються. Інтерпретація передбачає послідовне виконання інструкцій, без попереднього їх перетворення в код, прийнятний для процесора і операційної системи, як це робиться при компіляції. В .NET проміжна мова не інтерпретується, а компілюється в код при необхідності, так що не використовувані функції при цьому не будуть навіть скомпільовані. Відповідно, в C# немає та не може бути вбудованих інструментів, здатних в процесі роботи виконати сторонній програмний код, як це робиться, наприклад, в мові JavaScript за допомогою функції eval.

В даному випадку все, що буде виконуватися додатком має вже знаходитися в його програмному коді. Тож виникає необхідність додання до програми можливості розпізнати і виконати сторонній програмний код без безпосереднього перезапуску. Один з можливих варіантів вирішення цього завдання - застосування методів автоматичної генерації коду. Тобто, ще на етапі розробки, в додатку прописується шаблон, що містить базову структуру коду (наприклад, оголошення бібліотек), в який в подальшому буде включено тіло функції, введеної користувачем через інтерфейс і який буде запущений і виконаний вже в модифікованому вигляді.

При цьому написання програми необхідно виконати, враховуючи специфіку предметної області. Специфіка полягає в тому, що при вирішенні нелінійних рівнянь за допомогою метода хорд визначається наближене значення кореня рівняння ітераційним шляхом, причому на кожній ітерації програма працює з певною функцією, уведеною користувачем з клавіатури вже після запуску

програми. Оскільки інтегрувати цю функцію у вже запущений та працюючий код напряму неможливо, виникає необхідність динамічного запуску паралельного модуля, у який формула підставляється «на льоту».

Таким чином, послідовність роботи програми наступна. Користувач вводить рядок, що містить нелінійне рівняння, наближене значення кореня якого необхідно визначити. Після натиснення відповідної кнопки (рис. 1) відбувається динамічна генерація програмного коду на основі вже записаного в програмі шаблону (об'ява бібліотек та базова структура програмного коду), в який підставляється уведене користувачем рівняння, компіляція та виконання створеного додатку.

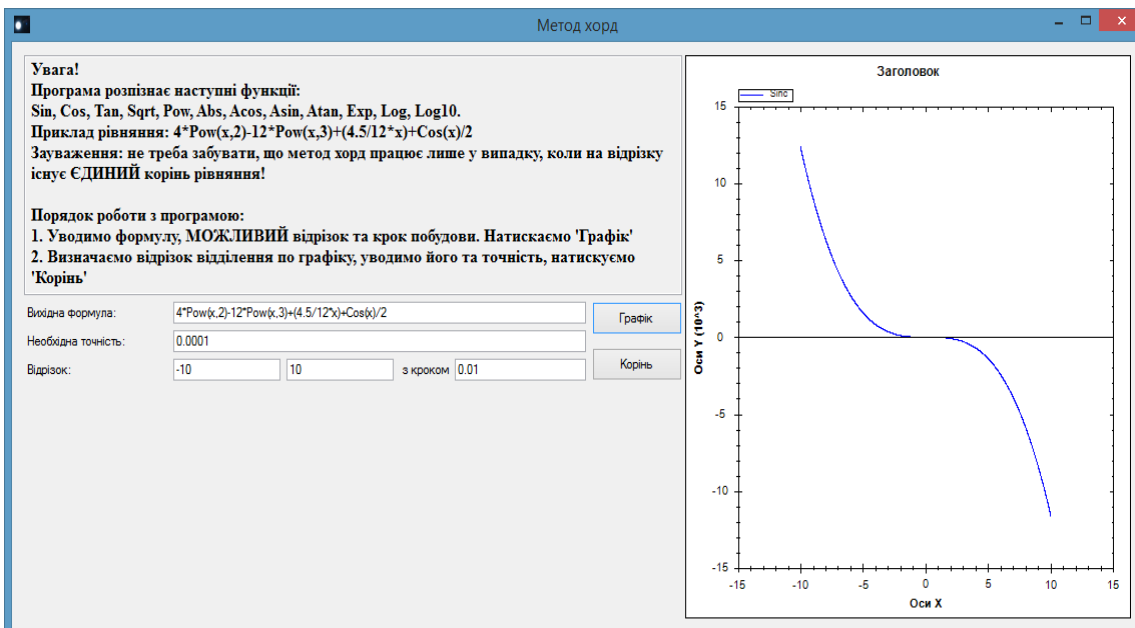


Рисунок 1 – Інтерфейс введення вхідних даних

Результати повертаються до основної програми у вигляді звіту з виконання функції виконання динамічно сформованого коду та виводяться на екран у зручному для користувача вигляді (рис. 2). Відповідно для цього необхідно, щоб функція, в тілі якої виконувалося обчислення кореня рівняння, повертала певне значення, тобто була типу, відмінного від void.

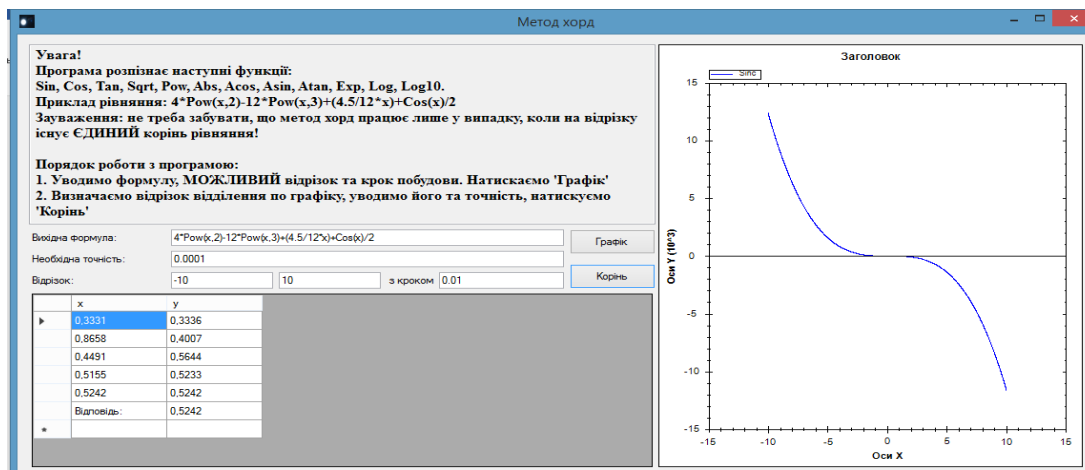


Рисунок 2 – Результат визначення кореня

Ключовим моментом є використання CodeDOM – складної сукупності класів, які забезпечують можливість використання динамічної генерації програмного коду [2, с 15]. CodeDOM містить типи, що представляють багато загальних типів елементів вихідного коду. За його допомогою можна розробити програму, яка буде будувати модель вихідного коду, використовуючи елементи CodeDOM для створення графа об'єктів. Цей об'єктний граф може бути відтворений у вигляді вихідного коду за допомогою вбудованого генератора коду для підтримуваного мови програмування. Вказаний інструмент можна також використовувати для компіляції вихідного коду в двійкову збірку.

У даному випадку вихідний код програми, що запускається динамічно, зберігається у окремому текстовому рядку. Ініціалізуючи новий об'єкт класу CompilerParameters (з зазначенням, що створювана програма буде запускатися лише в межах оперативної пам'яті, не маючи обміну інформацією з жорстким диском), отримуємо доступ компілятора C# в режимі реального часу, а через об'єкт класу CSharpCodeProvider маємо змогу отримати структуровані результати компіляції.

*Висновки.* Динамічна генерація коду - це прийом програмування, що полягає в тому, що фрагменти коду породжуються і запускаються безпосередньо під час виконання програми. Цей прийом був відомий досить давно, але ускладнення архітектури комп'ютерів, і, що особливо важливо, ускладнення наборів команд

процесорів привело до того, що в останні 10-15 років з'явилася можливість використання даного методу на більш високому рівні.

Метою динамічної генерації коду є використання інформації, доступної тільки під час виконання програми, для підвищення якості виконуваного коду. Іншими словами, динамічна генерація коду дозволяє спеціалізувати фрагменти програми за даними, які стають відомими лише під час виконання.

#### Література

1. *Троелсен Э.* Язык программирования C# 2005 и платформа .NET 2.0. 3-е издания / Э. Троелсен. – М.: Вильямс, 2007. – 795 с.
2. *Hazzard К.* Metaprogramming in .NET / К. Hazzard, J. Vock. – Manning, 2012. – 360 p.

### **ДИНАМИЧЕСКАЯ ГЕНЕРАЦИЯ КОДА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C # НА ПРИМЕРЕ РЕШЕНИЕ НЕЛИНЕЙНОГО УРАВНЕНИЯ МЕТОД ХОРД**

Кравченко В.А., Гавриленко Е.А.

***Аннотация*** – работа посвящена исследованию возможностей метода динамической (во время выполнения приложения) генерации и запуска программного кода на языке программирования C# на примере решения нелинейного уравнения методом хорд.

### **DYNAMIC GENERATE CODE IN THE PROGRAMMING LANGUAGE C# ON THE SOLUTION OF NONLINEAR EQUATIONS OF THE METHOD OF CHORDS**

V. Kravchenko, E. Gavrilenko

#### ***Summary***

**Work is devoted to the method of dynamic capabilities (at runtime) and start-up code in the programming language C # on the solution of the nonlinear equation by chords.**