

УДК 004.4'232

О. Овсяк, канд. техн. наук

*Львівська філія Київського національного університету культури і мистецтв,
Українська академія друкарства*

КЛАСИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ГЕНЕРУВАННЯ КОДУ

***Резюме.** Описано створену систему класів інформаційної системи комп'ютерного набору і редагування формул алгоритмів та генерування програмного коду з формул алгоритмів. Подано математичну модель створеної системи класів.*

***Ключові слова.** Модель, діаграма класів, операція секвентування, операція паралелення.*

A. Ovsyak

THE INFORMATIONAL SYSTEM GENERATING CODE CLASSES

***The summary.** The originated class system of computer typing and the algorithms formulae edition and program code generating from algorithms formulae are described. The mathematic model of created classes system are given.*

***Key words.** Model, classing diagrams, operation sequencing, operation paralleling.*

Постановка проблеми. Відомо 11 методів описання алгоритмів. Крім вербального та блок-схемного, є ще такі методи: віртуальних машин Поста (Posta) [1], Тюрінга (Turinga) [2], Ахо-Ульмана-Хопкрофта (Aho-Ullmana-Hopcrofta) [3], Шонгаге (Schönhage) [4], рекурсивних функцій (числення λ , Чорч (Church) [5], алгоритмів Маркова (Markova) [6], b-комплексів Колмогорова (Kolmogorova) (машина Kolmogorova) [7], універсальних алгоритмів Криніцького (Krinitskiego) [8] та алгебри алгоритмів [9]. Відомо [8, 10], що вербальним і блок-схемними методами, а також методами [1] – [8], алгоритми описують інтуїтивно, а не формально. Лише засобами алгебри алгоритмів отримують описання алгоритмів у вигляді формул – як на абстрактному, так і змістовному рівнях. Над формулами-алгоритмами, для їхньої мінімізації, на підставі властивостей операцій, можна виконати перетворення і дослідження, щоб виявити та усунути помилки, а також отримати належні результати. Власне, ці переваги алгебри алгоритмів над усіма іншими методами описання алгоритмів творять основу її застосувань для синтезу, мінімізації, дослідження математичних моделей і автоматизованого генерування програмного коду інформаційних технологій і систем з формул алгоритмів. Однак ще не створені комп'ютерні інформаційні системи, які б реалізували ці можливості алгебри алгоритмів.

Аналіз останніх досліджень і публікацій. Алгебра алгоритмів [9, 11, 13] утворена операціями секвентування, елімінування, паралелення, реверсування [14], рівності [14], операцій циклічних [9, 11 - 14] секвентування, елімінування і паралелення, які є новими і мають специфічні позначення. Причому геометричні розміри знаків операцій, крім знаків циклічних операцій, залежать від розмірів унітермів, над якими вони виконуються. Набір і редагування формул алгоритмів можливо здійснювати засобами відомих класичних інформаційних систем, до яких належать такі, як Word, Page Maker, Corel Draw, In Design та іншими. Виконувана при цьому значна кількість операцій робить використання традиційних інформаційних систем малоефективним. Для усунення цього недоліку створено спеціалізовану

комп'ютерну систему МОДАЛ [15]. Її засобами для заданих точок початку і кінця автоматично вирисовуються знаки операцій секвентування, елімінування і паралелення. В інформаційній системі АБСТРАКТАЛ [16] вже непотрібно задавати точки початку і кінця знаків операцій, а достатньо лишень задати місце розташування операції. Але системи МОДАЛ і АБСТРАКТАЛ мають свої спеціалізовані формати даних, які використовуються для записування формул алгоритмів та їхнього збереження і візуалізації, що утруднює створення трансляторів у випадку конвертації формул алгоритмів в інші інформаційні системи. Крім цього, у цих спеціалізованих системах немає доступу до баз алгоритмів, що суттєво утруднює використання попередніх напрацювань. Програмна реалізація цих систем на платформі Delphi значно ускладнює використання можливостей найсучаснішої і перспективної платформи Microsoft Visual Studio .NET для генерування програмного коду з формул алгоритмів.

Метою роботи є використання можливостей найсучаснішої платформи Microsoft Visual Studio .NET для реалізації математичної моделі системи класів інформаційної системи набору і редагування формул алгоритмів та, у перспективі, генерування з них кодів програм інформаційних систем.

Завдання дослідження. Створити й описати засобами алгебри алгоритмів модель системи класів, які реалізуються на платформі Microsoft Visual Studio .NET з використанням мов C#, XML і XAML [17].

Система класів формування унітермів і операцій. Унітерми алгебри алгоритмів можуть бути як текстовими, так і графічними об'єктами. Враховуючи ще й те, що знаки операцій є графічними об'єктами, для досягнення однотипності даних і спрощення коду доцільно розглядати унітерми і знаки операцій графічними об'єктами. У зв'язку з цим вводимо абстрактний клас (*Term*), який буде наслідуватися класами унітерма (*Uniterm*), операцій секвентування (*Sequence*), елімінування (*Elimination*), паралелення (*Parallelisation*), операцій циклічних секвентування (*CyclicSequence*), елімінування (*CyclicElimination*) і паралелення (*Cyclic Parallelisation*). Цю діаграму класів зображено на рис.1.

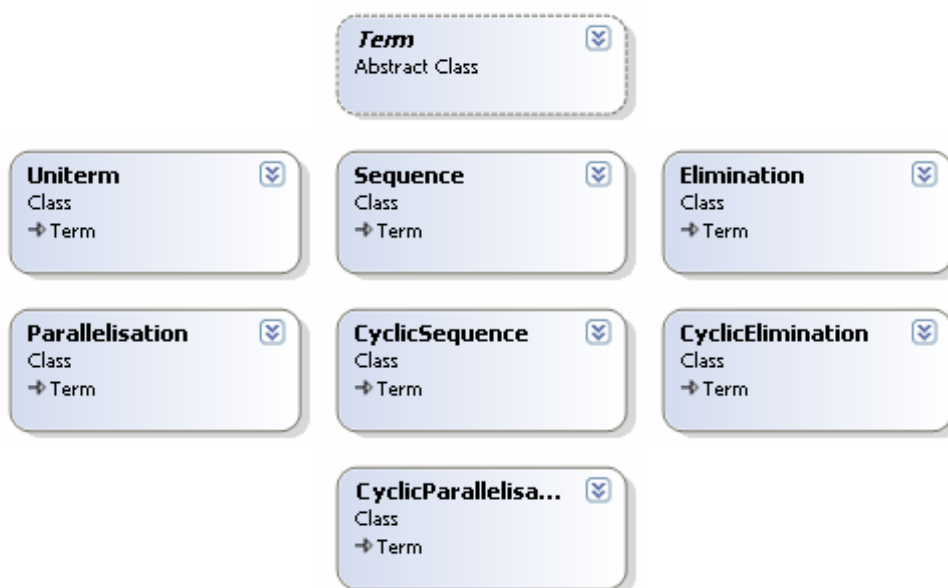


Рисунок1 – Абстрактний клас і його наслідники

Система незалежних класів. Створювана система генерування програмного коду може функціонувати поза операційними системами. Але використання операційної системи, завдяки вже готовим системним рішенням, суттєво спрощує проектування інформаційних систем. Найчастіше використовують операційні системи Windows, Solaris та інші. Враховуючи можливість створення на платформі Microsoft Visual Studio .NET, яка функціонує під керуванням операційної системи Windows, як самого редактора формул алгоритмів, так і бази алгоритмів, доцільним є використання операційної системи Windows.

Для здійснення зв'язків між інформаційною системою генерування коду та операційною системою вводимо спеціальний клас (*App*), див. рис. 2. Створення головного вікна інтерфейсу користувача системи генерування коду опишемо у класі головної форми (*MainForm*) (див. рис. 2).

Описання випадних вікон, у яких вибиратимуться орієнтації знаків операцій (горизонтальна, вертикальна) і розділювач між унітермами (кома, крапка з комою) створюють класи для операції секвентування – *SequenceForm*, операції елімінування – *EliminationForm* і операції паралелення – *ParallelisationForm* (див. рис. 2). Задання орієнтації (горизонтальна, вертикальна) циклічних операцій вводимо класи описування їхніх випадних вікон, а саме *CyclicSequenceForm* – для операції циклічного секвентування, *CyclicEliminationForm* – циклічного елімінування і *CyclicParallelisationForm* – циклічного паралелення.

Задання кеглів унітермів формул алгоритмів описується у спеціальному класі з назвою *FontForm*.

Клас *DeleteForm* вводиться для редагування формул алгоритмів, яке здійснюється заміною операції порожнім унітермом або першим чи другим унітермом операції з двома унітермами та секвентною областю циклічних операцій. Заміна вибраної формули – об'єкта новим або створення з нього першої чи другої гілки нової формули двоунітермових (секвентування, елімінування, паралелення) операцій описано у класі *InsertForm*, а одноунітермових циклічних операцій – у класі *InsertCyclicForm*.

Для задання параметрів (довжина, ширина, координати і т.д.) створюваних об'єктів (форм, панелей, кнопок і т.д.) графічних інтерфейсів користувачів уведено клас *FORM*. А класом *UnitermGraf* створюється випадне вікно для описування текстових унітермів формул алгоритмів.

MyCanvas і *MyDrawing* введено для ініціалізації та рисування графічних об'єктів формул алгоритмів на головному вікні.

Відображення у головному вікні створених спеціальних унітермів (знаки форми, панелі, кнопки і т.д.) виконується з використанням класу *SymvolAbstractalu*.

Алгоритми редагування циклічних операцій (заміни унітермом чи збереження підпорядкованої області) описано у класі *DeleteCyclicForm*.

Клас *WydalenniaFigur* призначено для витирання виділених фрагментів або цілої формули алгоритму.

DatabaseConfigForm – клас для створення і знищення баз алгоритмів, а клас *DatabaseForm* описує випадне вікно доступу до бази алгоритмів.

Клас створення з формули алгоритму програмного коду має назву *ZagalnyjGenKod*. Заміни об'єкта – формули або її фрагмента унітермом, секвентуванням, елімінуванням, паралеленням, циклічними операціями секвентуванням, елімінуванням і паралеленням містять алгоритми класу *ReplaceForm*.

Математична модель системи класів. Для побудови математичної моделі системи класів використовуємо розширену алгебру алгоритмів. Наслідування абстрактного класу *Term* класами-наслідниками *N* описуємо операцією секвентування з використанням розділювача крапки з комою, якою задається некомутативність унітермів, отримуємо таку формулу:



Рисунок 2 – Діаграма незалежних класів

$$\begin{pmatrix} Term \\ ; \\ N \end{pmatrix} \quad (1)$$

Класи-наслідники незалежні між собою, наявні всі одночасно, не впорядковані, тому для їхнього описання використовуємо операцію паралелення:

$$N = \left[\left[\left[Uniterm, Sequence \right], \left[Elimination, Parallelisation \right] \right], \left[\left[CyclicSequence, CyclicElimination \right], CyclicParallelisation \right] \right]. \quad (2)$$

Підставивши у формулу (1) значення (формулу (2)), отримуємо опис абстрактного класу і його наслідників у вигляді формули алгебри алгоритмів:

$$\left(\begin{array}{l} \text{Term} \\ ; \\ \left[\text{Uniterm, Sequence}, \text{Elimination, Parallelisation} \right] \\ , \\ \left[\text{CyclicSequence, CyclicElimination}, \text{CyclicParallelisation} \right] \end{array} \right) .$$

Незалежні класи (рис. 2) описуємо теж з використанням операції паралелення. Класи не є позиційованими, тому як розділювач використовуємо кому. Отримуємо в результаті таку формулу алгебри алгоритмів:

$$\left[\begin{array}{l} \left[\left[\text{App, MainForm}, \text{SequenceForm, EliminationForm} \right], \right. \\ \left. \left[\text{ParallelisationForm, CyclicSequenceForm} \right], \right. \\ \left[\left[\text{CyclicEliminationForm, CyclicParallelisationForm} \right], \right. \\ \left. \left[\text{FontForm, DeleteForm}, \text{InsertForm} \right], \right. \\ \left[\left[\left[\text{InsertCyclicForm, FORM}, \text{UnitermGraf} \right], \right. \right. \\ \left. \left. \left[\text{MyCanvas, MyDrawing}, \text{SymvolyAbstractalu} \right], \right. \right. \\ \left[\left[\left[\text{DeleteCyclicForm, WydalenniaFigur} \right], \right. \right. \\ \left. \left. \left[\text{DatabaseConfigForm, DatabaseForm} \right], \right. \right. \\ \left. \left. \left[\text{ZagalnyjGenKod, ReplaceForm} \right] \right. \right. \end{array} \right]$$

Результати дослідження

1. Створена діаграма, утворена абстрактним класом і його наслідниками, є базою для однотипного графічного представлення усіх складових формул алгоритмів, що спрощує реалізацію інформаційної системи.
2. Реалізація незалежних класів забезпечує редагування формул алгоритмів і генерування програмного коду.
3. Уперше засобами алгебри алгоритмів описано модель діаграми класів інформаційної системи. Базовий клас із класами-наслідниками описано некомутативною операцією секвентування, а незалежні класи – комутативним паралеленням.

Висновки

1. Створену модель системи класів можна програмно реалізувати мовою об'єктно-орієнтованого програмування C# на платформі Microsoft Visual Studio .NET версії 2008 року.
2. Програмна реалізація класів забезпечує підвищення рівня автоматизації процесів набору і редагування формул алгоритмів, описаних засобами алгебри алгоритмів.

Література

1. Post E.L. Finite Combinatory Processes - Formulation 1 / Post E.L // *Journal of Symbolic Logic*. – 1936. - № 1. – P. 103 – 105. (Reprinted in *The Undecidable*, pp. 289ff).
2. Turing A.M. On computable numbers, with an application to the Entscheidungsproblem / Turing A.M. // *Proceedings of London Mathematical Society*. – 1936 – 1937. – Series 2, vol. 42. – P. 230 – 265. (Correction, ibidem, vol. 43, pp. 544-546. Reprinted in [13 Davis M., pp. 155 – 222] and available online at <http://www.abelard.org/turpap2/tp2-ie.asp>).
3. Aho A.V., Hopcroft J.E., Ullman J.D. The design and analysis of computer algorithms / Aho A.V., Hopcroft J.E., Ullman J.D. // Addison-Wesley Publishing Company. – 1974.
4. Schönhage A. Universelle Turing Speicherung. In J. Dörr and G. Hotz, Editors / *Schönhage A. // Automatentheorie und Formale Sprachen, Bibliogr. Institut, Mannheim*. – 1970. – P. 369 – 383.
5. Church A. An unsolvable problem of elementary number theory / Church A. // *American Journal of Mathematics*. – 1936. – Vol. 58. – P. 345 – 363.
6. Markov A.A. Theory of algorithms (in Russian) / Markov A.A. // *Editions of Academy of Sciences of the USSR*. – 1951. – Vol. 38. – P. 176 – 189. (Translated into English in *American Mathematical Society Translations*, 1960, series 2, 15, pp. 1 – 14).
7. Kolmogorov A.N. On the concept of algorithm (in Russian). *Uspekhi Mat. Nauk* 8:4. – 1953. – P. 175 – 176. (Translated into English in *Uspensky V.A., Semenov A.L. Algorithms: Main Ideas and Applications*, Kluwer, 1993).
8. Krinitski N.A. Algorithms around us (in Russian) / Krinitski N.A. – Moscow: “Mir”, 1988. – 223. (Also translated to Spanish (*Algoritmos a nuestro alrededor*)).
9. Овсяк В. Засоби еквівалентних перетворень алгоритмів / В. Овсяк // *Доповіді національної академії наук України*. – 1996. – №9. – С. 83 – 89.
10. Успенский В. Теория алгоритмов: основные открытия и приложения. / В. Успенский, А. Семенов – М.: Наука, 1987. – 288 с.
11. Овсяк В. АЛГОРИТМИ: аналіз методів, алгебра впорядкувань, моделі, моделювання / Овсяк В. – Львів, 1996. – 132с.
12. Овсяк В. АЛГОРИТМИ: методи побудови, оптимізації, дослідження вірогідності / Овсяк В. – Львів: Світ, 2001. – 160 с.
13. Owsiak W., Owsiak A., Owsiak J. Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych / Owsiak W., Owsiak A., Owsiak J. – Opole: Politechnika Opolska, 2005. – 275 s.
14. Ovsyak V.K. Computation models and algebra of algorithms / Ovsyak V.K. // *Інформаційні системи та мережі. Вісник національного університету “Львівська політехніка”*. – 2008. – № 621. – С. 3– 18.
15. Бритковський В.М. Моделювання редактора формул секвенційних алгоритмів: автореф. дис. на здобуття наук. ступеня канд. тех. наук: спец. 01.05.02 “Математичне моделювання та обчислювальні методи” / В.М. Бритковський – Львів, 2003. – 18 с.
16. Василюк А.С. Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів: автореф. дис. на здобуття наук. ступеня канд. тех. наук: спец. 01.05.02 “Математичне та програмне забезпечення обчислювальних машин і систем” / А.С. Василюк – Львів, 2008. – 20 с.
17. Мак-Дональд М. WPF: Windows Presentation Foundation в NET 3.5 с примерами на C# 2008. Для профессионалов / Мак-Дональд М.; пер. с англ. Я.П. Волковой, Д.Я. Иваненко, Н.А. Мухана. – Москва, Санкт-Петербург, Киев: И.Д.Вильямс, 2008. – 928 с.

Одержано 02.11.2009 р.