

## ДОВГА АРИФМЕТИКА В EXCEL. II. РАЦІОНАЛЬНІ ЧИСЛА

В. Фурман, М. Хом'як, Я. Марко

*Львівський національний університет імені Івана Франка,  
вул. Грушевського, 4, 79005, Львів, Україна  
[fourman@i.ua](mailto:fourman@i.ua)*

Запропоновано VBA-реалізацію для Excel довгих арифметичних операцій для раціональних чисел з довільною розрядністю чисельника та знаменника. Із застосуванням об'єктно-орієнтованого підходу побудовано типовий інтерфейс класу SRational та описано його функціональність. Передбачено багатоваріантність символічного подання раціональних чисел, що структурно мають до трьох полів: ціла частина, чисельник та знаменник, а між ними – інтуїтивно зрозумілі розділювачі, зокрема, символи арифметичних операцій та дужки. Для обчислення арифметичних виразів використовуємо їхнє представлення як двійкових дерев. Проаналізовано функціональний комплекс для синтаксичного розбору раціональних чисел серед вхідної символічної інформації, а також імпорту даних з текстових файлів різних форматів.

*Ключові слова:* надбудова Excel, раціональні числа, алгоритм довільної точності, символічні обчислення, двійкове дерево.

У першій частині статті ми приділили увагу алгоритмічним та програмним особливостям реалізації точних обчислень над полем цілих чисел довільної розрядності в електронних таблицях Excel [1]. Запропонована VBA-реалізація є базою для конструювання раціональних чисел та складніших алгоритмів [2,3]. Мета цього дослідження – реалізувати клас раціональних чисел, а також зручні засоби для вводу та імпорту даних, передбачаючи їхню багатоваріантність.

**Інтерфейс класу раціональних чисел.** Мова VBA не цілком підтримує постулати об'єктно-орієнтованого програмування, а поняття "клас" реалізовано через "модуль класу" (Class Module). У модулі класу, на відміну від звичайних модулів, описані члени класу: методи (процедури й функції), властивості та дані. Запропонований далі інтерфейс класу раціональних чисел (SRational) цілком оригінальний і не претендує на роль стандарту [4,5,6], а слугує завданню продемонструвати основні ідеї використання дробів для точних обчислень, зокрема, для розв'язування систем лінійних алгебричних рівнянь. Клас SRational реалізує символічну арифметику в десятковій системі числення, що інтуїтивно зрозуміла для людини, хоча і не найефективніша з погляду реалізації [1].

Таблиця 1. Інтерфейс класу SRational – раціональних чисел у символічному представленні.

Група	Процедура / Функція / Властивість	Аргументи	Опис або приклади
процедури-конструктори	Init0, Init1	–	0 → 0/1; 1 → 1/1
	*** Init	m As String, Optional n As String = ""	"1", "2" → 1/2; "3" → 3/1
	*** Init3	sLeader As String, Optional sNum As String = "", Optional sDenom As String = ""	"1", "2", "3" → 1+2/3=5/3; "4", "", "" → 4/1
	InitBigInt	sNum As String	"2" → 2/1
	InitInt	i As Long	2 → 2/1
	*** InitLRatio	m As Long, n As Long	2,3 → 2/3
	*** InitDbI	s As String	"1.33e-3" → 133/10000
	*** Init_	v1() As Byte, n1_ As Integer, v2() As Byte, n2_ As Integer, iSignum As Integer	клонування: копіювання даних
перевірки стану (так/ні)	Перевіряє дріб $n/d$ , чи ...		
	IsBase_ IsInfinity IsUndefined	без аргументів	... $n, d \in \{1,0\}$ ? ... $n \neq 0, d = 0$ ? ... $n = d = 0$ ?
	IsZero		..., $n = 0, d \neq 0$ ?
	IsUnit, IsAbsUnit		... $n = \pm 1, d = 1$ ?
	IsPositive IsNegative		... $n > 0$ або $n < 0$ ?
	IsInteger		... $d = 1$ ?
	IsProperFraction		... $n < d$ ?
форматований друк	Numerator Denominator IntPart FractionalPart		без аргументів
	ToStr ToPaar ToTriplet ToNativeStr ToArithmExpr	5/3 → "5/3"; 5/3 → "(5;3)" 5/3 → "(1;2;3)" 5/3 → "1(2/3)" 5/3 → "1+2/3"	
	ToDbI	5/3 → 1,666...	
	StrFormat	Optional fFormat As Long = 0	

Таблиця 1. (Продовження)

Група	Процедура / Функція / Властивість	Аргументи	Опис або приклади
арифметичні операції та порівняння	Compare		повертає -1,0 або 1
	IsEqual		повертає так/ні
	*** Add, Subtract, Mult, Divide ArithmeticOp (арифм. дія за кодом)	y As SRational	повертає результат дії – посилання на об'єкт класу SRational
	*** Degry	n As Integer	
	*** X_YZ ( $x - y * z$ )	y, z As SRational	
	Simplify	–	спростити дріб: $4/6 \rightarrow 2/3$

*Примітка.* Символом "\*\*\*" помічено процедури та функції класу, що мають останній параметр виду "Optional bSimplify As Boolean = False". Для компактності у цій таблиці його не вказано. Якщо цей параметр задати явно як True, то обов'язково запуститься процедура спрощення дробу Simplify.

Інтерфейс класу SRational представлений такими групами процедур, функцій та властивостей: 1) процедури-конструктори, 2) перевірка значень, 3) реалізація арифметичних операцій та спрощення дробів, 4) вивід у символному форматі (Табл. 1). Загалом, функціональність членів класу зрозуміла з їхніх англійських назв. Коротко проаналізуємо кожен з груп.

VBA підтримує тільки конструктор без параметрів, тому оператор виду "Set obj = New SRational" створює неініційований об'єкт класу. Спосіб ініціалізації, коли задано чисельник та знаменник можна вважати звичайним для раціональних дробів. Тут символний формат – це послідовність цифр, взагалі кажучи, довільної довжини з діапазону Integer. Ми також передбачили ініціалізацію з трьома числами-стрічками, де виокремлено цілу частину (перша стрічка), а друга та третя – чисельник і знаменник дробової частини. Ціле число  $k$  може бути представлено як раціональне ( $k/1$ ), але для нього зручніше мати окремий конструктор. Можливі й інші конструктори. Зручно мати конструктори з вбудованих типів даних, таких як ціле Long або Double (із заданою кількістю знаків після коми або в стрічковому форматі, в загальному випадку).

Відмітимо, що оператор присвоєння одного об'єкта іншому (наприклад, Set obj2 = obj) насправді збільшує лічильник посилань, а для клонування (створення копії внутрішніх даних) треба мати окремий метод, що копіює дані для чисельника та знаменника – Init\_. Необов'язковий параметр процедур ініціалізації – логічне значення "Так" (True) вказує на необхідність спрощення виразу-дробу (за замовчуванням спрощення – Simplify – не виконується, див. також примітку після Табл. 1).

Друга група – властивості для перевірки значень. Вони мають доступ "тільки читання" (Property Get) і повертають логічне значення "Ні" або "Так" (False/True). Вони можуть використовуватися, зокрема, для організації розгалужень у складних алгоритмах, наприклад, для врахування нульових елементів розріджених матриць.

Група функцій порівняння та арифметичних операцій реалізують основну, "корисну" функціональність класу. Ці функції повертають новий об'єкт класу, а початкові об'єкти (аргументи операції) не змінюються. Зауважимо, що в об'єктно-орієнтованих мовах програмування, таких як C++ або C#, вони реалізовані, як правило, як переважання операторів [4,5] (наприклад, оператор VBA "Set x=x.Add(y)" еквівалентний "x+=y").

Нарешті, для видруку раціональних чисел у символному (текстовому) форматі клас SRational надає низку функцій ToString, ToPair, ToTriplet, ToNativeStr, ToArithmExpr, які пропонують різні зручні формати. Той же результат дає єдина функція StrFormat з аргументом, що здійснює селекцію одного з форматів (див. Табл. 1).

**Формати вхідних та вихідних даних.** Зручність та багатоманітність способів задавання вхідної інформації є одним з визначальних факторів успіху програмного забезпечення, що реалізує довгу арифметику та обчислення з використанням раціональних чисел. Джерелами вхідних даних для програм-макросів в електронних таблицях є комірки (діапазони комірок) на одному чи декількох аркушах, що зберігаються у файлах (статичні дані) або дані, створення деякою підпрограмою, які використовуються далі іншою програмою (динамічні дані). Також вірогідно припустити, що досить великі масиви даних (наприклад, матриці) можуть бути сформовані незалежно іншими програмами і збережені, наприклад, у текстових файлах згідно з деякими правилами, згідно з визначеним форматом даних (зовнішні дані).

Щоб програма Excel розпізнавала дані як свої, раціональні числа в комірках таблиць зберігатимуться у текстовому форматі, а під час виконання макросів вони "на льоту" мають бути перетворені до об'єктів класу SRational. Останнім етапом виконання кожного з макросів має бути обернене перетворення – видрук і збереження в заданих комірках об'єктів SRational як текстових стрічок (за допомогою функції StrFormat, див. Табл.1).

**Організація обчислень і операції вводу-виводу.** Орієнтуючись на матричні обчислення, можемо вважати, що результати перед збереженням у комірках мають бути оформлені як масиви стрічок з двома індексами, і кожен елемент масиву буде збережено у відповідній комірці відносно виділеного або вказаного діапазону комірок. Якщо використовується механізм "формула масиву", то спочатку потрібно виділити діапазон клітинок і ввести формулу-функцію, аргументами якої є символні стрічки – адреси одного або більше діапазонів, що містять раціональні числа. Після натиснення комбінації клавіш Ctrl+Shift+Enter результати (текстові стрічки з погляду програми Excel) будуть автоматично збережені в комірках (клас Range) згідно із загальними правилами: якщо виділений діапазон менший за масив результатів, то не всі дані будуть збережені, а якщо він більший, то частина комірок міститиме признак помилки на зразок "#Н/Д" (немає даних). Інший механізм, який реалізують процедури-макроси (наприклад, через натиснення командних кнопок у вікнах діалогу) передбачає, що мають бути задані, як аргументи, вхідні та вихідні діапазони комірок. Оскільки доступ до решти комірок фактично не обмежено, то, наприклад, для вихідних даних достатньо вказати верхній

лівий кут (ВЛК), відносно якого зберігатимуться результати. Куди і які результати будуть записані, – має реалізувати програмна логіка макроса-процедури. Перевага механізму функцій – у можливості автоматичного перерахунку формул, що одночасно може бути й недоліком, якщо перерахунок достатньо тривалий ("підвисання" програми) [7]. У тілі функції також блоковано доступ на запис у будь-яку комірку. Навпаки, макроси-процедури, хоча й не передбачають автоматичного перерахунку, але мають повний доступ (читання і запис) як до комірок XLS(x)-файла, так й до інших файлів, що дає змогу виконувати імпортно-експортні операції зі зовнішніми даними.

На рис. 1 показано діалогове вікно для імпортування масивів раціональних чисел (матриць та векторів) із текстових файлів деяких поширених форматів (Matlab, Maple, з розділювачом-пробілом чи табуляцією тощо). Приклад такого файлу – на рис. 2. Передбачено функціональність автоматичного розпізнавання формату файлів, а також різні стрічкові формати збереження раціональних чисел у комірках.

Функції видруку масивів раціональних чисел у комірки мають забезпечити їхню сумісність з вхідними форматами, з метою повторного (подальшого) використання. Однак інтуїтивно зрозумілі та інваріантні щодо результуючого значення записи раціональних чисел мають більшу різноманітність, можуть включати, наприклад, додаткові пробіли та дужки, як от: " $1 + (2/3)$ ", " $-(2 - 2/3)$ ", " $-1 (2/3)$ " чи навіть вирази, такі як: " $-(1 + 2/3) / 2$ ", " $(2/3)^3$ " або " $-.2e1 / 1.3e1$ ". Можна розділити їх на три види: 1) три або два поля з цілими числами (можливо, зі знаком) і з розділювачами полів, якими можуть бути, зокрема, знаки арифметичних операцій; 2) десяткові числа з плаваючою крапкою та, можливо, показником степеня; 3) арифметичні вирази. У табл. 2 наведено деякі приклади записів раціональних чисел, які слугували тестовим набором під час відладки.

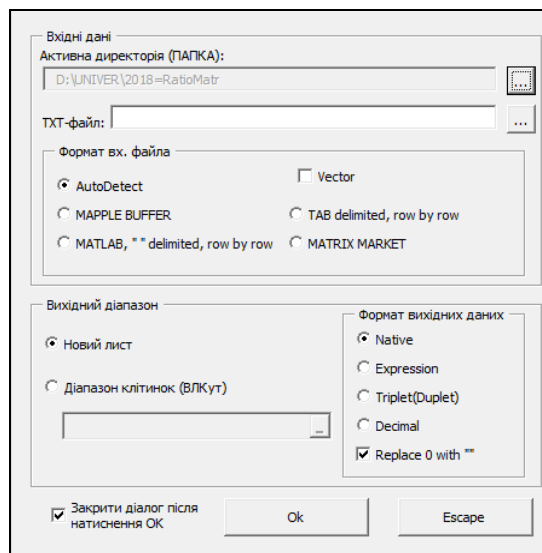


Рис. 1. Діалогове вікно імпорту даних з деяких текстових файлів та їхнього збереження у комірках електронних таблиць.





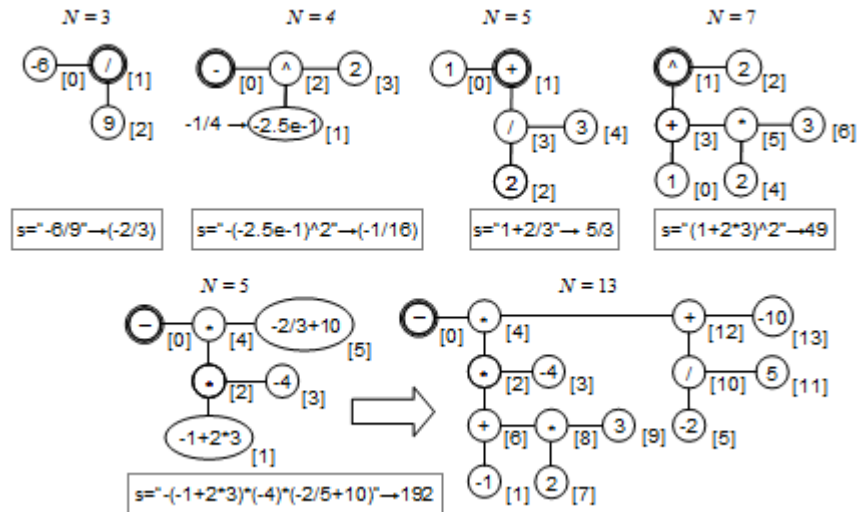


Рис. 3. Деякі арифметичні вирази та відповідні їм двійкові графи. Над графами показано кількість вершин-вузлів  $N$ , а під ними в рамці – символічний вираз та його результат. Два нижніх графи демонструють розгортання виразів у дужках (показано еліпсом) у нові підграфи.

Функція `Str2Decimals` здійснює аналіз чисел з плаваючою крапкою, виділяє показник степеня та коректує положення розділювача між цілою й дробовою частиною, після чого вхідна стрічка ділиться на дві – цілу частину та чисельник дробу, а знаменник формується як символічна стрічка з нулів та з одиницею ліворуч відповідної довжини, згідно з розрядністю чисельника (без нулів праворуч).

Найгроміздкіша в реалізації – функція `Str2IntAndFractionS`, що "розпізнає" різні формати подання раціональних чисел (структурно можуть бути – з двома або трьома полями), зокрема, й ті, які використовує функція друку `StrFormat`, тобто операції вводу-виводу узгоджено, що логічно і зрозуміло. Якщо вважати знаки арифметичних операцій розділювачами полів, то ця функція `Str2IntAndFractionS` є альтернативою до використання класу `ArithmExpr` для найпростіших арифметичних виразів. Наприклад, вираз `"1+2/3"` або `"1+(2/3)"` має три поля, що розділені знаками операцій `"+"` і `"/"`, але вже вираз `"(1+2)/3"` функція `Str2IntAndFractionS` не розпізнає. Складність алгоритму зумовлює необхідність налаштування деяких опцій щодо розділювачів полів, знаків операцій, дужок тощо. Коротко, на нижньому рівні після ініціалізації налаштувань викликається функція `Read3SFields`, що може "читати" до трьох полів з допомогою викликів ще однієї функції `Read1Field`. По ходу виконання задіяно службові функції (утиліти) для роботи з текстом, що згруповані в модулі `SFieldUtil` (рис. 4).

Зацікавлений читач має змогу самостійно дослідити код, що доступний як електронний ресурс [[http://old.geology.lnu.edu.ua/phis\\_geo/nauka/metod/metod.htm](http://old.geology.lnu.edu.ua/phis_geo/nauka/metod/metod.htm)].



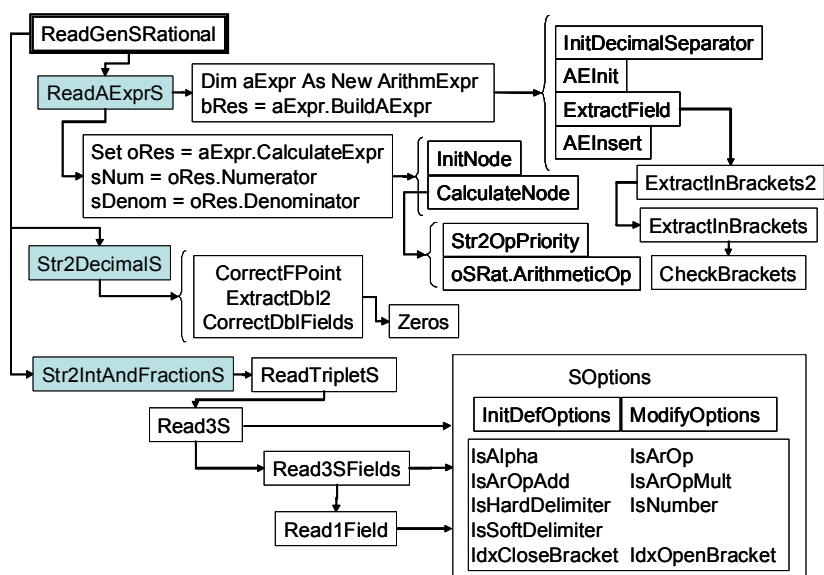


Рис. 4. Спрощена схема викликів функцій для опрацювання тестових стрічок – джерела даних для створення об'єктів класу раціональних чисел SRational.

Таке розгалуження способів подання символьних раціональних чисел для визначення об'єктів класу SRational є перевагою даної реалізації, оскільки відкриває додаткові зручності для користувача, що, на нашу думку, важливо для роботи в електронних таблицях.

#### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Фурман В. Довга арифметика в Excel. I. VBA-реалізація / В. Фурман, М. Хом'як, Я. Марко // Електроніка та інформаційні технології. – 2019. – Вип. 11. – С. 81–98.
2. Knuth D. E. The Art of Computer Programming, 1. Fundamental Algorithms, 3rd Ed. – Addison-Wesley. –1997. – 650 p.
3. Knuth D. E. The Art of Computer Programming, 2. Seminumerical Algorithms, 3rd Ed. – Addison-Wesley. –1998. – 762 p.
4. Rational [Electronic recourse] .– Mode of access: <https://ruby-doc.org/core-2.4.0/Rational.html>.
5. The Rational Class [Electronic recourse] .– Mode of access: [https://www.math.utah.edu/docs/info/libg++\\_21.html](https://www.math.utah.edu/docs/info/libg++_21.html)
6. Rational.py: An example Python class [Electronic recourse] .– Mode of access: <http://infohost.nmt.edu/tcc/help/lang/python/examples/rational/>
7. Excel performance: Improving calculation performance [Electronic recourse].– Mode of access: <https://docs.microsoft.com/en-us/office/vba/excel/concepts/excel-performance/excel-improving-calculation-performance>.
8. Cormen T.H. Introduction to Algorithms, 3rd Ed. / T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. – MIT Press, 2009. – 1312 p.

---

**LONG ARITHMETIC IN EXCEL. II. RATIONAL NUMBERS****V. Fourman, M. Khomyak, Ya. Marko**

*Ivan Franko National University of Lviv,  
Hrushevskogo Str. 4, UA-79005 Lviv, Ukraine,  
fourman@i.ua*

Within VBA and Excel framework the arithmetic operations for rational numbers with arbitrary digits in the fraction "numerator per denominator" are implemented. Symbolic calculations work with their representation in electronic spreadsheet cells as text and full precision integer algorithms. Using the object-oriented approach, a typical interface of the class SRational is constructed and its functionality is described. Class methods can be grouped as constructors, arithmetic operators and functions to print a rational number back into a symbolic string. Constructors convert symbols into internal representations as arrays of digits. The result of some operation can be returned to the user as one of the intuitive formats, in particular, as a fraction with a separated integer part, or as a triple of numbers, or as an arithmetic expression. Also a routine to simplify the fraction "on demand" is implemented, which is not always necessary in the intermediate calculations.

There is provided a multivariate symbolic representation of rational numbers structurally having up to three fields: an integer, a numerator, and a denominator. Between them can be intuitive separators, in particular, symbols of arithmetic operations and brackets. To calculate arithmetic expressions, we use their representations as binary trees. The complex of the functionality to parse rational numbers in the symbolic input information is analyzed.

Instead of manual input of large rational numbers, you can use already formed arrays. Therefore, data import from text files of various formats is important. It provides interaction with other programs.

*Key words:* EXCEL Add-In, rational numbers, full precision algorithm, symbolic computation, binary tree.

*Стаття: надійшла до редколегії 24.02.2019*

*доопрацьована 05.05.2019*

*прийнята до друку 25.05.2019*