

УДК [378.147+373.5]:004.7

DOI <https://doi.org/10.32782/cusu-pmtp-2023-2-5>

ПАРСЕР АРИФМЕТИЧНИХ ВИРАЗІВ ДЛЯ ВИВЧЕННЯ ОСНОВ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

Лупан Ірина Володимирівна,

кандидат педагогічних наук, доцент,

доцент кафедри інформатики та інформаційних технологій

Центральноукраїнського державного університету імені Володимира Винниченка

ORCID ID: 0000-0002-4791-0445

Гацелюк Сергій Володимирович,

бакалавр факультету математики, природничих наук та технологій

Центральноукраїнського державного університету імені Володимира Винниченка

ORCID ID: 0009-0000-0695-5164

Нові покоління мов програмування потребують внесення в методiku навчання нових ідей та понять, без знання яких студент не зможе проявити на практиці стійких навичок написання реальних комп'ютерних програм. Методика навчання дисциплін, пов'язаних з програмуванням, вимагає ретельного опрацювання дидактичного матеріалу, представлення його у різноманітних формах для кращого усвідомлення та засвоєння студентами. Особливо це стосується методики навчання паралельних обчислень. Якісне засвоєння пов'язаних з ним понять, технологічних та алгоритмічних підходів можна забезпечити, якщо на перших етапах використовувати наочні моделі паралельних процесів. Прикладом такої моделі може бути схема обчислення арифметичного виразу, який містить арифметичні оператори та дужки.

Оскільки окремі дії, відповідно до пріоритетів арифметичних операцій, можна виконувати одночасно, то розпаралелювання такого виразу e , з одного боку, задачею досить простою для початкового ознайомлення з паралельними обчисленнями, а з іншого, дозволяє виділити окремі підзадачі, встановити зв'язки між ними, визначити характеристики потенційного паралельного алгоритму, дослідити залежність між кількістю наявних обчислювачів (процесорів) та часом виконання алгоритму тощо.

У статті описано програмний засіб *TreeBuilder*, розроблений в рамках дослідження, для візуалізації дерева формули, побудованої за введеним арифметичним виразом, та обчислення характеристик алгоритму обчислень. Засіб *TreeBuilder* може бути корисним для формування «паралельного мислення» у студентів молодших курсів та старшокласників.

Для побудови дерева формули та обчислення характеристик алгоритму обчислень в рамках дослідження було розроблено алгоритм парсингу арифметичних виразів та здійснено його програмну реалізацію мовою C#.

Ключові слова: арифметичний вираз, парсер, паралельний алгоритм, навчання паралельного програмування.

Lupan Iryna, Gatseliuk Serhii. The arithmetic parser for teaching the basics of parallel computing

New generations of programming languages need to introduce new ideas and concepts to teaching, without which the student will not be able to show in practice stable skills to write real computer programs. The methodology of teaching programming disciplines requires careful processing of didactic material, presenting it in various forms for better awareness and learning of students. This is especially true of the methods of teaching parallel computing. Qualitative learning, related concepts, technological and algorithmic approaches, can be ensured by using visual models of parallel processes in the first stages. An example of such a model is the scheme of calculating the arithmetic expression that contains arithmetic operators and brackets.

Since separate actions, in accordance with the priorities of arithmetic operations, can be computed at the same time, the parallelization of such an expression is, on the one hand, the task is quite simple to familiarize with parallel calculations, and, on the other hand, allows you to identify separate tasks, determine the characteristics of a potential parallel algorithm, investigate the relation between the number of computers (processors) and the time of execution of the algorithm, etc.

The article describes the TreeBuilder software designed to visualize the formula tree, built by the entered arithmetic expression, and to calculate the characteristics of the computing algorithm. The Treebuilder can be used for the formation of “parallel thinking” in students and high school pupils.

To build a formula tree and to calculate the characteristics of the algorithm of computing within the investigation an algorithm of parsing arithmetic expressions was developed and its software implementation was made in the C#.

Key words: *arithmetic expression, parser, parallel algorithm, training in parallel programming.*

Вступ. «Паралельні та розподілені обчислення» як окрема галузь були включені до Computing Curricula ще у редакції 2013 року [1]. Відповідно, така дисципліна є обов’язковою складовою професійної підготовки фахівців спеціальності «Комп’ютерні науки» [2]. Однак проблема формування паралельного мислення, методики навчання паралельного програмування, формування відповідних фахових компетентностей залишається актуальною.

Проблемам формування змісту навчальної дисципліни «Паралельні та розподілені обчислення», проектування візуального середовища та відбору засобів для навчання студентів паралельного програмування присвячено публікації Ю.О. Сіциліцина [3; 4; 5], І.В. Крашеніннік [6], О.С. Литвина [7] та інших. Між тим на першому етапі ознайомлення студентів, а згодом, можливо, і учнів профільних класів закладів загальної середньої освіти, важливо сформувані у них «паралельне мислення», тобто здатність ділити задачу на підзадачі, розпаралелювати потоки даних, якими будуть обмінюватися підзадачі, що виконуються паралельно, уявляти зв’язки між підзадачами тощо. На цьому етапі важливо надати студенту/учню засіб, який би дозволив представити наочно процес розпаралелювання, поекспериментувати з моделлю задачі, дослідити її поведінку за різних умов, з різними параметрами.

Найпростішим прикладом такої моделі для паралельних процесів може бути арифметичний вираз: при виконанні послідовних обчислень (тобто одним виконавцем) порядок виконання дій визначається пріоритетами арифметичних операцій. Якщо ж виконавців декілька, то вони можуть виконувати деякі дії одночасно, за потреби обмінюючись їхніми результатами.

Матеріали та методи. Інтерфейс розробленого в рамках дослідження програмного додатка TreeBuilder побудований на базі Windows Forms – технології для розробки десктопних програм мовою програмування C# – об’єктно-орієнтованої мови програмування, що використовується для розробки програмного забезпечення на платформі Microsoft. Як редактор коду було використано Visual Studio – інтегроване середовище розробки (IDE) від компанії Microsoft, призначене для розробки програмного забезпечення для платформ Windows, Android, iOS, web-додатків та інших.

Для побудови дерева формули та виконання необхідних обчислень в рамках дипломної роботи був розроблений та реалізований алгоритм парсингу арифметичних виразів.

Результати. Програмний додаток TreeBuilder призначений для моделювання розпаралелювання арифметичних виразів, які містять односимвольні операнди, арифметичні оператори (“+”, “-”, “*”, “/”) та дужки. Порівняно з першою версією додатка [8], у другій було змінено інтерфейс, доопрацьовано алгоритми парсингу виразів і побудови дерева формули, що дозволило усунути недоліки попередньої версії.

У новій версії TreeBuilder може запропонувати користувачу декілька сценаріїв роботи. Перший сценарій («Help») – виклик інструкції, в якій вказано, які коди клавіш застосунок приймає, яких дій слід уникати, якого порядку дій слід дотримуватись, щоб отримати результати для свого виразу.

Наступний сценарій – це введення виразу в поле «Expression». Крім того, за бажанням можна ввести числове значення – кількість процесорів – в поле «Processors». Після цього, натиснувши кнопку «Build Tree», можна отримати графічне зображення дерева виразу, а натиснувши кнопку «Characteristics», отримати результати підрахунків: кількість операндів виразу, кількість ярусів (паралельних кроків), кількість необхідних для обчислення процесорів (за умови необмеженого

паралелізму), а також ефективності і прискорення потенційного паралельного алгоритму обчислення введеного виразу.

Отже, алгоритм застосунку складається з двох головних модулів – «Побудова дерева» та «Характеристики дерева», які складаються з підчасти. Блок-схема алгоритму «в крупних командах» представлена на рисунку 1.

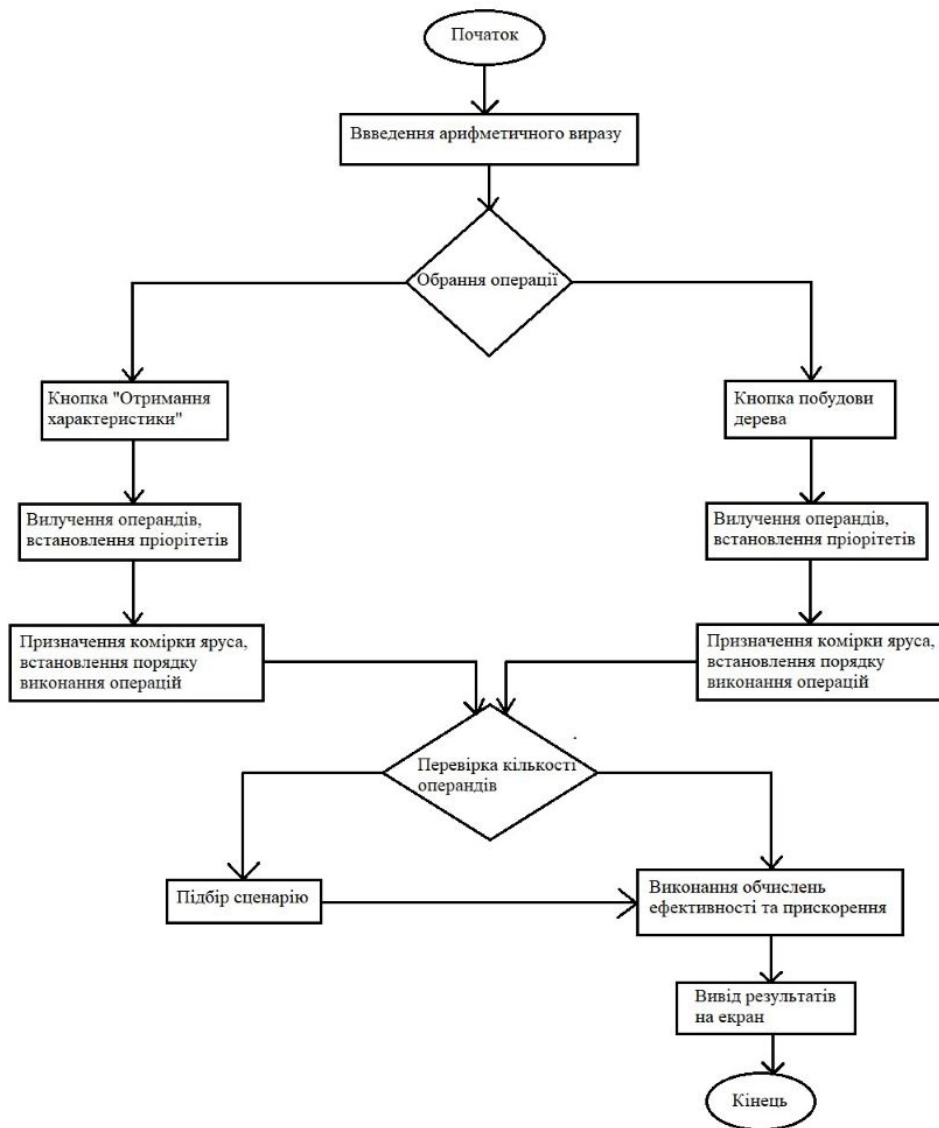


Рис. 1. Блок-схема алгоритму

Призначення першого модуля полягає в тому, щоб за введеним виразом виконати обчислення координат з'єднувальних ліній, операндів та операторів для презентації на екран дерева формули відповідно до введеної користувачем кількості процесорів, керуючись при цьому загальними правилами пріоритету виконання арифметичних операцій.

Для обчислення координат потрібні такі дані: кількість літер, кількість операндів, пріоритетність кожного операнда та кількість процесорів (якщо користувач не запропонував своє значення, або ввів неправильно).

Призначення другого модуля полягає в обчисленні характеристик можливого паралельного алгоритма для введеного арифметичного виразу. В ході виконання цього блока алгоритму за кількістю процесорів, кількістю операндів та ярусів дерева формули обчислюються прискорення та ефективність.

Прискорення (*SpeedUp*) паралельного алгоритму обчислюють за формулою $S_p(n) = \frac{T_1(n)}{T_p(n)}$, де $T_1(n)$ – час виконання послідовного алгоритму. Якщо вважати, що час виконання всіх операцій однаковий, то за $T_1(n)$ можна взяти кількість операцій (*Operands*). $T_p(n)$ – час виконання паралельного алгоритму на p процесорах, тобто кількість ярусів (*Floors*).

Ефективність паралельного алгоритму визначається формулою $E_p(n) = \frac{S_p(n)}{p}$.

Потрібна кількість обчислювачів (або процесорів) p визначається як максимальна ширина яруса дерева обчислень.

Чим ближче значення $S_p(n)$ до p , а $E_p(n)$ – до одиниці, тим «кращим» є побудований паралельний алгоритм.

На рисунку 2 показано результати обчислення вказаних характеристик для виразу 1: “ $a+a+a+a+a+a+a+a+a+a+a+a+a+a+a$ ”, а на рисунку 3 – побудоване дерево формули. Як бачимо, дерево відповідає класичній схемі здвоєння, а кількість процесорів визначається відповідно до моделі необмеженого паралелізму.

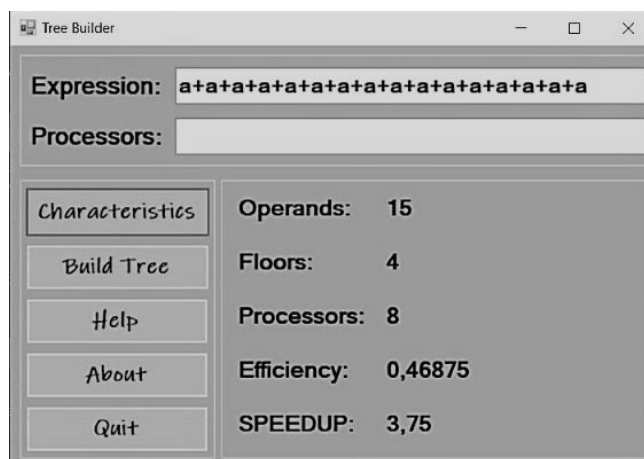


Рис. 2. Результати обчислень для виразу 1

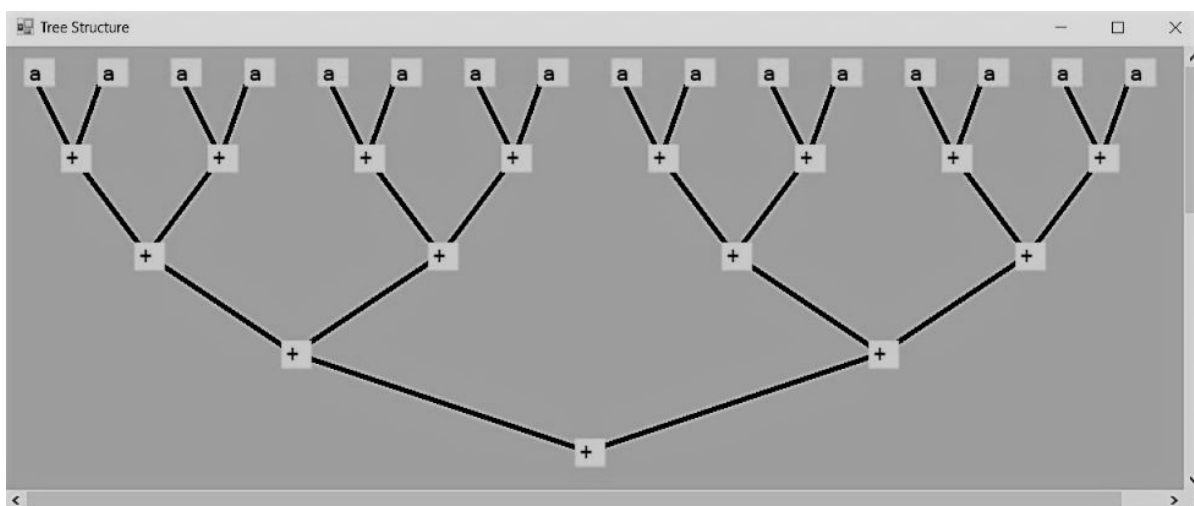


Рис. 3. Дерево формули для виразу 1

За допомогою додатка TreeBuilder можна провести дослідження залежності прискорення та ефективності паралельного алгоритму від кількості процесорів, а також перевірити виконання принципу Брента, тобто з’ясувати, як збільшиться час виконання паралельного алгоритму при зменшенні кількості процесорів.

перетворення виразу (чого додаток TreeBuilder поки не робить), однак застосування такого додатка дозволяє експериментувати з різними виразами та досліджувати змінення характеристик паралельного алгоритму значно швидше і наочніше.

Висновки. Розробка програмних засобів, які дозволяють унаочнити процес розпаралелювання задач, дозволить зосередити увагу студентів саме на паралельному програмуванні, показати його переваги, звернути увагу на проблеми, які виникають при виконанні паралельних програм тощо. Звичайно, моделювання тільки лиш обчислення арифметичних виразів для навчання студентів паралельних обчислень недостатньо. Однак завдання формування у студентів «паралельного мислення», подолання стереотипів, вироблених в результаті написання «послідовних» програм, потребує ефективних рішень, одним із яких є створення візуальних засобів і середовищ, застосування яких значно покращить якість вивчення дисципліни «Паралельні обчислення».

Література:

1. Computer Science Curricula 2013 Curriculum Guidelines for Undergraduate Degree Programs in Computer Science December 20, 2013 The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society DOI: 10.1145/2534860. URL: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.
2. Стандарт вищої освіти України першого (бакалаврського) рівня ступеня «бакалавр» за галуззю знань 12 «Інформаційні технології» спеціальністю 122 «Комп’ютерні науки» : МОН України, 2019. URL: <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2019/07/12/122-kompyut.nauk.bakalavr-1.pdf>.
3. Сіциліцин Ю.О. Формування змісту професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень. *Педагогічні науки: теорія та практика*. 2022. № 2(42). С. 105–111. doi.org/10.26661/2786-5622-2022-2-16.
4. Сіциліцин Ю.О. Моделювання змісту дисципліни «паралельні та розподілені обчислення». *Педагогічні науки: теорія та практика*. 2023. Vol. 4(44). С. 22–28. doi.org/10.26661/2786-5622-2022-4-03.
5. Сіциліцин Ю.О. Проектування візуального середовища для навчання студентів паралельного програмування. *Вісник Запорізького національного університету. Педагогічні науки, Частина II*. 2020. № 3(26). С. 116–120. doi.org/10.26661/2522-4360-2020-3-2-17.
6. Крашеніннік І.В. Паралельні і розподілені обчислення у структурі фахової підготовки майбутніх інженерів-програмістів зі скороченим терміном навчання. *Інформаційні технології в освіті та науці: зб. наук. пр.* 2017. Т. 1. № 9. С. 144–147.
7. Литвин О.С. Вивчення основ паралельних та розподілених обчислень із використанням одноплатних мікрокомп’ютерів Raspberry Pi. *Інноваційні технології в освіті* : зб. матеріалів Міжнародної науково-технічної конференції, 9-11 квіт. 2019. м. Івано-Франківськ, 2019. С. 127.
8. Гацелюк С.В. Графічна програма-парсер для арифметичних виразів. *НАУКОВІ ЗАПИСКИ МОЛОДИХ УЧЕНИХ*, 2022. № 10. URL: <https://phm.cuspu.edu.ua/ojs/index.php/SNYS/article/view/1965>.

References:

1. Computer Science Curricula 2013 Curriculum Guidelines for Undergraduate Degree Programs in Computer Science December 20, 2013 The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society DOI: 10.1145/2534860 – URL: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf
2. Ministry of Education and Science of Ukraine (2019). Standart vyshchoi osvity Ukrainy pershoho (bakalavrskoho) rivnia stupenia “bakalavr” za haluzziu znan 12 “Informatsiini tekhnolohii” spetsialnistiu 122 “Kompiuterni nauky” [The standard of higher education of Ukraine of the first (bachelor) level of the “bachelor” degree in the field of knowledge 12 “Information technologies” specialty 122 “Computer science”]. Kyiv: Ministry of Education and Science of Ukraine. Retrieved from: <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2019/07/12/122-kompyut.nauk.bakalavr-1.pdf> [in Ukrainian].
3. Sitsylitsyn Yu. O. (2022) Formuvannia zmistu profesiinoi pidhotovky maibutnix inzheneriv-prohramistiv u haluzi paralelnykh obchyslen [Formation Of The Content Of Professional Training of Future Software Engineers In The Field Of Parallel Computing], *Pedahohichni nauky: teoriia ta praktyka*, issue 2 (42), pp. 105-111. doi.org/10.26661/2786-5622-2022-2-16 [in Ukrainian].

4. Sitsylitsyn Yu. O. (2023) Modeliuvannya zmistu dystsypliny «paralelni ta rozpodileni obchyslennia». [Modeling Of The Contents Of The Discipline “Parallel And Distributed Computing”]. *Pedahohichni nauky: teoriia ta praktyka*, issue 4 (44), pp. 22–28. doi.org/10.26661/2786-5622-2022-4-03 [in Ukrainian].
5. Sitsylitsyn Yu. O. (2020) Proiektuvannya vizualnoho seredovyshcha dlia navchannia studentiv paralelnoho prohramuvannia [Designing Visual Environment For Training Students Parallel Programming]. *Visnyk Zaporizkoho natsionalnoho universytetu. Pedahohichni nauky, Part II*, № 3 (26), pp. 116–120. doi.org/10.26661/2522-4360-2020-3-2-17 [in Ukrainian]
6. Krasheninnik I. V. (2017) Paralelni i rozpodileni obchyslennia u strukturі fakhovoi pidhotovky maibutnikh inzheneriv-prohramistiv zi skorochenym terminom navchannia [Parallel And Distributed Computing In The Structure Of Professional Training Of Future Software Engineers With A Shortened Period Of Study]. *Informatsiini tekhnolohii v osviti ta nauksi: zb. nauk. pr. [Information technologies in education and science: Proceedings of Scientific papers]*. V. 1, issue 9, pp. 144–147 [in Ukrainian]
7. Lytvyn O. S. (2019) Vyvchennia osnov paralelnykh ta rozpodilenykh obchyslen iz vykorystanniam odnoplatsnykh mikrokompiuteriv Paspberry Pi. [Learning the basics of parallel and distributed computing using Raspberry Pi single-board microcomputers] *Innovatsiini tekhnolohii v osviti : zb. materialiv Mizhnarodnoi naukovo-tekhnichnoi konferentsii [Proceedings of the International Scientific and Technical Conference]*. Ivano-Frankivsk [in Ukrainian]
8. Hatseliuk S. V. (2022) Hrafichna prohrama-parser dlia aryfmetychnykh vyraziv. [The Program For A Graphical Parser Of Arithmetic Expressions]. *Naukovi zapysky molodykh uchenykh*, № 10. Retrieved from <https://phm.cuspu.edu.ua/ojs/index.php/SNYS/article/view/1965> [in Ukrainian].