

РОЗВ'ЯЗАННЯ МАТЕМАТИЧНИХ ЗАДАЧ ЗАСОБАМИ МОВИ ПРОГРАМУВАННЯ PYTHON

*Осадча К.П., Хромішев О.В.
Мелітопольський державний педагогічний університет
імені Богдана Хмельницького, м. Мелітополь
e-mail: okp@mdp.u.org.ua*

Постановка проблеми в загальному вигляді. Нині для математичних розрахунків існує багатий набір різноманітних засобів і програмного забезпечення персональних комп'ютерів (у тому числі планшетних комп'ютерів і смартфонів), а саме: калькулятори (Калькулятор Windows, The Calc), системи комп'ютерної математики (Mathematica, Maple), статистичне програмне забезпечення (SPSS, Statistica), розв'язувачі конкретних математичних задач (Solver, UMS), мови програмування для математичних розрахунків (Fortran, R, APL) та мови програмування загального призначення (Pascal, C, Python,). Чисельні розрахунки потребують багато часу, тому на сучасному етапі розвитку обчислювальної техніки актуальним постає питання розробки ефективних програм розв'язання математичних задач, які швидко здійснюють обчислення.

Аналіз останніх досліджень і публікацій. Особливості мови програмування Python розглядали Бізлі Д., Гифт Н., Джонс Д., Круглик В.С., Лутц М., Саммерфілд М., Сваруп С.Х., Сузі Р.А. та ін. Практичними розробками для мови програмування Python займалися Пилєва О.О., Хахаєв І.А., зокрема засоби математичних обчислень Python вивчали Кадан А.М., Репін А.Ю. Соловйов І.А., Червяков А.В., Шібзухов З.М. та ін. Зокрема Кадан А. М. зазначає, що Python є популярною мовою програмування у академічному середовищі західних університетах через його різноманітні обчислювальні аспекти [1]. Репін А.Ю. Соловйов І.А., Червяков А.В. відзначають таку властивість цієї мови програмування як те, що незалежно від платформи Python можна використовувати як потужний інженерний калькулятор завдяки с стандартному потужному математичному модулю math [2, с. 17]. Шібзухов З.М. вважає, що Python, маючи зручний для читання синтаксис, доцільно використовувати для розв'язання математичних задач, тому що він має інструментарій для наукових обчислень й інженерних розрахунків, візуалізації моделей і даних, доступний на будь-якій платформі (Windows, Linux, OS X і т.д.) [3].

Виділення невирішених раніше частин загальної проблеми, яким присвячується стаття. Проте науковцями і IT-спеціалістами не достатньо

висвітлено питання ефективності окремих засобів мови програмування Python для розв'язання типових математичних задач.

Постановка завдання. Отже, актуальним завданням постає з'ясування можливостей і засобів математичних обчислень мови програмування Python та виділення серед них найбільш ефективного засобу для вирішення різних типів математичних задач.

Виклад основного матеріалу дослідження з повним обґрунтування отриманих наукових результатів. Python є вільно розповсюджуваною мовою програмування з великою кількістю вільно розповсюджуваних модулів і бібліотек для створення програм майже будь-якого типу (веб-додатки, робота з базами даних, наукові обчислення, інженерні розрахунки, візуалізація даних, розробка користувальницького інтерфейсу і т.д.). Вважається, що Python неважко вивчити, через що можна за нетривалий час почати використовувати його для вирішення поставлених задач. Python широко застосовується, починаючи з Google, NASA і кінчаючи студіями по створенню анімаційних фільмів, має простий і зручний для читання синтаксис [3]. Ця мова близька до MATLAB і зручна для програмування математичних обчислень. До того ж Python вміє працювати з такими мовами як Fortran, C, C++, які вже широко використовуються у розрахунках.

Можна виділити такі математичні можливості мови програмування Python: 1) існує чотири вбудованих типу числових символів: булеві значення, цілі числа, числа з плаваючою точкою, комплексні числа; 2) підтримуються всі основні математичні операції і дії; 3) наявні математичні модулі: decimal, fractions, math, numbers, random та ін.; 4) наявні бібліотеки для складних обчислень та візуалізації даних: SymPy, Numeric, NumPy, Pandas, Matplotlib, Dislin та ін. [8].

Серед величезної кількості завдань, що зустрічаються в обчислювальній техніці, можна виділити кілька типів, яким вчені завжди приділяли особливу увагу. Такий інтерес викликає або практичним значенням задачі, або якимись цінними її властивостями, що представляють особливий інтерес для дослідження. Найбільш важливі типи задач виділяє Левітін А.: сортування; пошук; задачі з теорії графів; комбінаторні задачі; геометричні задачі; чисельні задачі [4, с.45].

Для досягнення мети дослідження нами було обрано три типові задачі: задача на знаходження найбільшого спільного дільника (НСД), задача на обчислення факторіалу числа, задача на обчислення послідовності Фібоначі. Вони відносяться до задач комбінаторного та чисельного типів, тому розглянемо їх докладніше.

Бондаренко М.Ф. комбінаторними називає задачі, для вирішення випадків підрахунку кількості можливих комбінацій об'єктів, які задовольняють певним властивостям. Це задачі на перелічення, задачі про існування та побудову, задачі про вибір умов, за яких можна здійснити вибір підмножини або деякої сукупності частин множини так, щоб задовольнити певні вимоги [5, с.48–49]. Завдяки тому, що Python дозволяє виражати

алгоритми коротко і просто, вирішення комбінаторних задач засобами цієї мови програмування реалізується більш ефективно порівняно з іншими мовами програмування.

Як зазначає Левітін А.В. чисельні задачі мають справу з математичними об'єктами, які за своєю суттю є безперервними. Прикладами типових чисельних завдань є: рішення рівнянь і систем рівнянь, обчислення визначених інтегралів і значень функцій і т.д. [4, 51]. Перевагою Python над іншими мовами програмування при розв'язанні чисельних задач є вбудована підтримка «довгої арифметики», комплексних чисел, списків, словників, стеків і черг, що позбавляє помилок при обрахуванні дробових чисел, які у комп'ютері можуть бути представлені тільки з певною похибкою.

Для розв'язання задачі на знаходження НСД ми використали такі засоби мови програмування: умовні оператори, функції та цикл з умовою, рекурсія. Для кожного з цих засобів написано було окрему програму. Для оцінки ефективності запропонованих засобів для розв'язання задачі на знаходження НСД засобами Python ми використали елементи модуля time, які допомагають дізнатися, як довго виконується програма на мові Python. Як середовище для запуску програм було обрано платформу Anaconda, яка є безкоштовним дистрибутивом мови програмування Python для великомасштабної обробки даних, прогнозного аналізу і наукових обчислень [7].

Для обчислення НДС у всіх 3-х програмах нами було введено такі вхідні дані: 125 і 75. У підсумку виконання програм ми отримали такі результати: програма з використанням умовного оператора та циклу з передумовою виконувалася 5,929 с, програма з використанням функції та циклу з передумовою – 4,601 с, програма з використанням рекурсії – 3,779 с. Отже, можна зробити висновок, що найбільш ефективним засобом для наведеної задачі є використання рекурсії.

Для розв'язання задачі на обчислення факторіалу числа ми використали такі засоби мови програмування Python: умовний оператор і цикл з умовою, функції з рекурсією, лише функції, lambda-функцію, спеціальний оператор модулю math. Для обчислення у всіх 5-ти програмах нами було введено такі вхідні дані: 3000. У підсумку виконання програм ми отримали такі результати: програма з використанням умовного оператора і циклу з передумовою виконувалася 0,001000 с, з використанням функції з рекурсією – 0,012 с, з використанням лише функцій – 0,026 с, з використанням lambda-функції – 0,001001 с, з використанням спеціального оператора модулю math – 0,004 с. Отже, можна зробити висновок, що найбільш ефективним засобом для цієї задачі є використання у програмі умовного оператора і циклу з умовою.

Для розв'язання задачі на обчислення послідовності Фібоначі ми використали такі засоби мови програмування Python: функція, що включає цикл з передумовою, разом із циклом з умовою, об'єктно-орієнтоване програмування, функція і цикл з умовою. У підсумку виконання 3-х програм ми отримали такі результати: програма побудована на використанні функції з

циклом з передумовою і циклу з умовою задіяла 0,0008 с процесорного часу, програма з використанням об'єктно-орієнтованого програмування – 0,0005 с, програма з використанням функції з циклом з умовою – 0,0018 с. Таким чином, можна зробити висновок про те, що найбільш ефективним засобом для розв'язання цієї задачі є використання засобів об'єктно-орієнтованого програмування мови Python.

Висновки з даного дослідження і перспективи подальших кроків у даному напрямі. У результаті проведеного дослідження можна зробити такі висновки: ефективним засобом для швидкого розв'язання задачі на знаходження найбільшого спільного дільника є використання рекурсії; для задачі на обчислення факторіалу числа – використання у програмі умовного оператора і циклу з умовою; для задачі на обчислення послідовності Фібоначі – засоби об'єктно-орієнтованого програмування мови Python. Таким чином, на нашу думку, швидкість математичних обчислень, а отже й ефективність програм мовою Python, залежить від типу вирішуваної задачі і мовних конструкцій, що використовуються при розробці програмного коду. Так використання рекурсії у другій задачі не показало такої ефективності як при вирішенні першої задачі. Отже, цей засіб не може бути ефективним для розв'язання всіх математичних задач. Проте, щоб зробити такий висновок доцільно провести подібні експерименти над розв'язанням інших типів і різновидів математичних задач. Також у перспективах нашого дослідження заплановано з'ясування ефективності стандартних і додаткових модулів для математичних та інженерних розрахунків.

Література

1. Кадан А. М. К вопросу использования языка Python при изучении математических дисциплин студентами ИТ-специальностей. – [Електронний ресурс]. – Режим доступу : URL: <http://elib.bsu.by/bitstream/123456789/104337/1/Кадан-192.pdf>.
2. Соловьёв И.А. Вычислительная математика на смартфонах, коммуникаторах и ноутбуках с использованием программных сред Python: учеб. пособ. / И.А. Соловьёв, А.В. Червяков, А.Ю. Репин. – СПб.: Издательство «Лань», 2011. – 272 с.
3. Основы вычислительной аналитики на Python. Спецкурс профессора Шибзухова З.М. – [Електронний ресурс]. – Режим доступу : URL: <http://tidm.ru/osnovy-vychislitelnoy-analitiki-na-python>.
4. Левитин А.В. Алгоритмы: введение в разработку и анализ: пер. с англ. / Левитин А.В.– М.: Издательский дом «Вильямс», 2006. – 576 с.
5. Бондаренко М.Ф. Комп'ютерна дискретна математика: Підручник / М.Ф. Бондаренко, Н.В. Білоус, А.Г. Руткас. – Харків: «Компанія СМІТ», 2004. – 480 с.
6. Учебник Python 3.1. – [Електронний ресурс]. – Режим доступу : URL: https://ru.wikibooks.org/wiki/Python/Учебник_Python_3.1.
7. Why Anaconda? – [Електронний ресурс]. – Режим доступу : URL: <https://www.continuum.io/why-anaconda>.

8. Лутц М. Изучаем Python, 4-е издание / М. Лутц. – Пер. с англ. – СПб. : Символ-Плюс, 2011. – 1280 с.

Анотація. У статті здійснено аналіз математичних можливостей мови програмування Python. На прикладі розв’язання трьох типових математичних задач (на знаходження найбільшого спільного дільника, на обчислення факторіалу числа, на обчислення послідовності Фібоначі) засобами цієї мови програмування здійснено спробу доведення ефективності окремих засобів розв’язання цих задач, а саме: рекурсії, умовного оператору і циклу з умовою, об’єктно-орієнтованого програмування. Зроблено висновок про те, що швидкість математичних обчислень (ефективність програм мовою Python) залежить від типу вирішуваної задачі і мовних конструкцій, що використовуються при розробці програмного коду.

Ключові слова: математичні задачі, мова програмування, Python, ефективність програм.

Аннотация. В статье проведен анализ математических возможностей языка программирования Python. На примере решения трех типовых математических задач (нахождение наибольшего общего делителя, на вычисление факториала числа, на вычисление последовательности Фибоначчи) средствами этого языка программирования предпринята попытка доказательства эффективности отдельных средств их решения, а именно: рекурсии, условного оператора и цикла с условием, объектно-ориентированного программирования. Сделан вывод о том, что скорость математических вычислений (эффективность программ на языке Python) зависит от типа решаемой задачи и языковых конструкций, используемых при разработке программного кода.

Ключевые слова: математические задачи, язык программирования, Python, эффективность программ.

Abstract. The article analyzes the mathematical possibilities of the Python programming language. As an example of solving of three typical mathematical tasks (to find the greatest common divisor, the number factorial, Fibonacci sequence evaluation) using the means of this programming language it is taken a shot to prove the efficiency of separate means to solve these tasks. Especially: recursion, conditional operator and cycle with the condition, object oriented programming. It is concluded that the rate of mathematical calculations (the efficiency of the programs written in Python) depends on the type of task and language constructs, used in development of program code.

Keywords: mathematical tasks, programming language, Python, programs efficiency.