

УДК 004.076.2

*Царева О.С., асистент,*

*Цап Д. І., здобувач*

*Івано-Франківський національний технічний університет нафти і газу*

## **ЕНЕРГОЕФЕКТИВНІСТЬ ПРОГРАМНОГО КОДУ: АНАЛІЗ СУЧАСНИХ ТЕНДЕНЦІЙ**

Енергоефективність програмного коду сьогодні виходить на передній план серед глобальних викликів ІТ-індустрії, перетворюючись із глибокої концепції на критичну необхідність. Із значним зростанням обсягів даних та обчислювальних потужностей екологічний слід цифрових технологій стає дедалі відчутнішим. Екологічне програмування — це екологічно стійка обчислювальна практика, яка прагне мінімізувати енергію, задіяну в обробці рядків коду, і, у свою чергу, допомогти організаціям зменшити загальне споживання енергії.[1] Ця тенденція зумовлює появу нового напрямку, мета якого полягає у мінімізації вуглецевого сліду на всіх етапах розробки. Оптимізація коду здатна не лише знизити навантаження на довкілля, а й суттєво зменшити операційні витрати компаній на утримання хмарної інфраструктури.

Однією з ключових сучасних тенденцій є свідомий вибір інструментів розробки та архітектурних шаблонів з огляду на їхнє енергоспоживання. Дослідження доводять, що компільовані мови програмування є більш енергоефективними за інтерпретовані, що змушує інженерів шукати баланс між швидкістю написання коду та ресурсомісткістю готового продукту. В середньому програми на C і Rust є найбільш енергоефективними. Програми на Java споживають приблизно в 2 рази більше енергії, ніж програми на C. Програми на JavaScript/TypeScript споживають від 4 до 20 разів більше енергії, ніж програми на C. Нарешті, програми на Python є чорними вівцями з енергоспоживанням у 75 разів більшим, ніж програми на C.[2] Це означає перехід від традиційного підходу, орієнтованого виключно на швидкість виконання функцій, до архітектури, яка ретельно враховує показники використання процесора та оперативної пам'яті.

На рівні безпосереднього написання коду тенденції зводяться до суворого усунення надмірності та ефективного використання апаратних ресурсів. Важливим аспектом стає мінімізація передачі даних через мережу, агрегація запитів та продумане кешування на клієнтському боці. Оптимізація програмного коду є ключовим кроком у досягненні енергоефективності, високої продуктивності та цілей сталого розвитку. Екологічний вплив виконання неефективного коду, хоча і не завжди очевидний, збільшує вуглецевий слід компанії, яка розробляє

неефективний код.[3] Видалення неактивного (мертвого) коду, зниження алгоритмічної складності для уникнення непотрібних вкладених циклів та відмова від постійного опитування серверів стають фундаментальною основою екологічного програмування. Таким чином, енергоефективність сьогодні формує нове розуміння чистого коду, вимагаючи від фахівців усвідомлення того, як програмні рішення впливають на фізичний світ.

Окрему увагу варто приділити ролі центрів обробки даних (data centers), які є основою сучасної цифрової інфраструктури. За оцінками міжнародних досліджень, дата-центри споживають значну частку світової електроенергії, а їхній внесок у глобальні викиди CO<sub>2</sub> постійно зростає. Неefективний програмний код прямо впливає на навантаження серверів, збільшуючи потребу в охолодженні, електроенергії та додатковому обладнанні. Таким чином, навіть незначна оптимізація програм може мати масштабний екологічний ефект у глобальному вимірі.

Таким чином, енергоефективність сьогодні формує нове розуміння чистого коду, вимагаючи від фахівців усвідомлення того, як програмні рішення впливають на фізичний світ. З огляду на це, екологічне програмування стає не просто технічною перевагою, а професійною етикою. Впровадження інструментів моніторингу енергоспоживання у CI/CD процеси та вибір алгоритмів з меншою обчислювальною складністю дозволяють розробникам створювати продукти, які є економічно вигідними для бізнесу та безпечними для навколишнього середовища. Врешті-решт, справжня якість коду в епоху кліматичних викликів вимірюється не лише відсутністю помилок, а й мінімальною кількістю ват-годин, витрачених на його виконання.

#### **Список використаних джерел:**

1. Ibm Cloud Education Team. How environmentally friendly organizations can use green coding to drive long-term success. *IBM. Topics*. URL: <https://www.ibm.com/think/topics/green-coding> (дата звернення: 07.03.2026).
2. Julien Richard-Foy, Wojciech Mazur. Sustainable Scala. *Scala. Blog*. URL: <https://www.scala-lang.org/blog/2021/12/14/sustainable-scala.html> (дата звернення: 07.03.2026).
3. CGI IT UK Ltd, University of Leicester. Green software guide. URL: [https://www.cgi.com/sites/default/files/2025-02/green\\_software\\_guide.pdf](https://www.cgi.com/sites/default/files/2025-02/green_software_guide.pdf) (дата звернення: 07.03.2026).