

## ЕФЕКТИВНІСТЬ МЕТОДІВ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ І ПАТЕРНІВ ДЛЯ ПЛАНУВАННЯ СІЛЬСЬКИХ РЯТУВАЛЬНИХ ФОРМУВАНЬ

*Анатолій Тригуба<sup>1</sup>, д. т. н., Лілія Коваль<sup>2</sup>, ад'юнкт, Андрій Татомир<sup>1</sup>, к. т. н., Олег Боярчук<sup>1</sup>, к. т. н., Роман Падюка<sup>1</sup>, к. т. н., Володимир Станько<sup>1</sup>, к. е. н.*

*<sup>1</sup> Львівський національний університет природокористування, вул. Володимира Великого, 1, м. Дубляни, Львівський р-н, Львівська обл., Україна,*

*<sup>2</sup> Львівський державний університет безпеки життєдіяльності, вул. Клепарівська, 35, м. Львів, Львівська обл., Україна,*

*e-mail: trianamik@gmail.com; kovallilia494@gmail.com; andrew.tatomyr@gmail.com; padyukari@lnup.edu.ua; VStanko@ukr.net*

<https://doi.org/10.32718/agroengineering2025.29.208-217>

### **Тригуба А., Коваль Л., Татомир А., Боярчук О., Падюка Р., Станько В. Ефективність методів об'єктно-орієнтованого програмування і патернів для планування сільських рятувальних формувань**

У статті обґрунтовано підхід до підвищення ефективності програмних моделей, призначених для планування добровільних рятувальних формувань у сільських громадах, на основі інтеграції методів об'єктно-орієнтованого програмування та патернів проектування. Показано, що поєднання принципів SOLID із використанням патернів Strategy та Observer забезпечує суттєве зменшення архітектурної складності та створює передумови для розроблення масштабованих і гнучких рішень, які здатні адаптуватися до зміни алгоритмів і зовнішніх умов. Реалізовані програмні модулі мовою Python 3.12 дозволили провести експериментальне оцінювання впливу підходів програмування на ключові показники якості системи. Дослідження виконано на прикладі завдань планування розташування добровільних рятувальних формувань для Шептицької міської територіальної громади Львівської області. Результати доводять, що використання класичних процедурних підходів є недостатньо ефективним, оскільки такі системи характеризуються низькою гнучкістю, значними витратами на супровід і слабкими можливостями повторного використання коду. Перехід до класичного об'єктно-орієнтованого програмування (ООП) суттєво покращує показники, однак максимальний ефект забезпечує інтеграція ООП з патернами, що дозволило підвищити якість реалізації до 0.92, масштабованість – до 0.89, повторне використання компонентів – до 0.91 та водночас знизити складність архітектури до 0.40. Це підтверджує доцільність впровадження патернів для побудови сучасних інформаційних систем у сфері безпеки та планування рятувальної інфраструктури в сільських громадах. Практична значущість полягає в можливості подальшої інтеграції розроблених модулів у системи просторового моделювання й оптимізації рятувальних формувань. Перспективи досліджень передбачають розширення набору алгоритмів оптимізації, застосування стохастичних і робастних методів, а також використання динамічних просторових даних для формування комплексного інструментарію підтримки управлінських рішень для громад.

**Ключові слова:** об'єктно-орієнтоване програмування, патерни проектування, ефективність, сільські громади, рятувальні формування, інформаційні системи, оптимізація.

### **Tryhuba A., Koval L., Tatomyr A., Boiarchuk O., Padiuka R., Stanko V. Effectiveness of object-oriented programming methods and patterns for planning rural rescue teams**

The article substantiates an approach to improving the effectiveness of software models designed for planning voluntary rescue teams in rural communities based on the integration of object-oriented programming methods and design patterns. It is shown that combining SOLID principles with the use of Strategy and Observer patterns significantly reduces architectural complexity and creates the conditions for developing scalable and flexible solutions that can adapt to changes in algorithms and external conditions. The software modules implemented in Python 3.12 allowed for an experimental assessment of the impact of programming approaches on key system quality indicators. The research was carried out using the example of tasks for planning the location of volunteer rescue teams for Sheptytskyi urban territorial community in Lviv region. The results prove that the use of classical procedural approaches is not effective enough, since such systems are characterized by low flexibility, significant maintenance costs, and poor code reuse capabilities. The transition to classic object-oriented programming (OOP) significantly improves performance, but the maximum effect is achieved by integrating OOP with patterns, which has increased the quality of implementation to 0.92, scalability to 0.89, component reuse to 0.91, and at the same time reduced the complexity of the architecture to 0.40. This confirms the feasibility of implementing patterns for building modern information systems in the field of security and planning rescue infrastructure in rural communities. The practical significance lies in the possibility of further integration of the developed modules into systems for spatial modeling

and optimization of rescue formations. Research prospects include expanding the set of optimization algorithms, applying stochastic and robust methods, and using dynamic spatial data to form a comprehensive set of tools to support management decisions for communities.

**Keywords:** object-oriented programming, design patterns, efficiency, rural communities, rescue teams, information systems, optimization.

**Постановка проблеми.** Сучасні виклики, пов'язані з безпекою сільських громад, зумовлюють необхідність пошуку нових підходів до організації та планування діяльності добровільних рятувальних формувань. В умовах воєнних дій, техногенних аварій та природних катастроф особливого значення набуває питання підвищення готовності місцевих громад до оперативного реагування на надзвичайні ситуації. Традиційні методи планування рятувальних структур ґрунтуються переважно на статистичних даних та експертних оцінках, що в багатьох випадках не забезпечує достатньої адаптивності й гнучкості системи [3]. У таких умовах застосування інформаційних технологій, зокрема об'єктно-орієнтованого програмування та патернів проєктування, відкриває можливості для створення ефективних програмних засобів моделювання та прогнозування потреби у формуванні рятувальних структур у сільських громадах.

Значна частина досліджень у галузі програмної інженерії акцентує увагу на важливості застосування патернів як універсальних рішень для проєктування гнучких і масштабованих програмних систем. Вони забезпечують повторне використання коду, структурованість архітектури та зменшення ризику помилок при розробці складних систем [10]. У поєднанні з методами об'єктно-орієнтованого програмування це дозволяє створювати інструменти підтримки ухвалення рішень, здатні враховувати просторові, демографічні та інфраструктурні характеристики сільських територій. Таким чином, поєднання методів програмної інженерії з моделями цивільного захисту дає змогу побудувати інноваційний підхід до планування рятувальних формувань.

Окремим викликом є інтеграція таких рішень у практичну діяльність місцевих органів влади та громадських організацій. Важливою стає можливість створення гнучких моделей, які можна адаптувати до конкретних умов території, враховуючи як густоту населення, так і доступність транспортної інфраструктури чи близькість критичних об'єктів. Сучасні дослідження у сфері громадської безпеки наголошують на тому, що цифрові інструменти для підтримки ухвалення рішень мають значно підвищувати стійкість громад і сприяти зменшенню наслідків надзвичайних

ситуацій [19]. Це обґрунтовує актуальність дослідження ефективності методів об'єктно-орієнтованого програмування та патернів для розробки програмних систем планування добровільних рятувальних формувань.

У цій статті запропоновано підхід до планування добровільних рятувальних формувань у сільських громадах на основі використання методів об'єктно-орієнтованого програмування та патернів проєктування. Запропонована концепція дозволяє створити адаптивні програмні моделі, які враховують особливості інфраструктури та демографічної ситуації на місцях, забезпечуючи підвищення точності прогнозування потреби у формуваннях та оптимізацію їхнього розташування. Таким чином, робота поєднує результати досліджень у сфері програмної інженерії та практичні завдання цивільного захисту, спрямовані на підвищення стійкості та безпеки сільських громад.

**Аналіз останніх досліджень і публікацій.** Науково-прикладне завдання, яке стосується планування добровільних рятувальних формувань у сільських громадах, перебуває на перетині двох напрямів, які потребують використання класичних моделей розташування об'єктів та сучасної програмної інженерії. У дослідженнях з управління ризиками і цивільного захисту послідовно підкреслюється, що локальна спроможність громад залежить від завчасного планування сил реагування, логістики та стійкості інфраструктури [12], а також від залучення добровольців і коректної інтеграції їх роботи у формальні процедури реагування [17]. Для сільських територій із розрідженою мережею доріг та низькою густотою населення це ставить особливі вимоги до просторової оптимізації розташування рятувальних підрозділів і пунктів базування.

Методологічну основу просторового планування задають моделі розташування-покриття й наближеного оптимального розташування: р-медіана, задача розташування з максимальним покриттям (MCLP), задача мінімального покриття (LSCP) та їхні розширення з часовими й імовірнісними обмеженнями [2; 8; 9; 13]. Узагальнений виклад цих підходів і сучасні алгоритмічні напрацювання наведено в монографіях Даскіна та оглядах Мюррея, де показано зв'язок між якістю

сервісу, часом доїзду та конфігурацією мережі [1.10; 1.11]. У прикладних роботах для служб екстреної допомоги доведено чутливість рішень до специфіки дорожньої мережі, сезонності трафіку та багатокритеріальних компромісів між охопленням і навантаженням на формування [2; 1.11]. Водночас більшість цих праць концентрується на оптимізаційних моделях і обмежено розкриває, як саме програмна архітектура систем підтримки ухвалення рішень впливає на відтворюваність розрахунків, масштабованість і адаптивність до нових даних.

У програмній інженерії ключову роль відіграють об'єктно-орієнтовані підходи та патерни проектування, що забезпечують повторне використання рішень, низьку зв'язаність компонентів і чітке відокремлення відповідальностей. Базові ідеї зафіксовано у класичних працях «Design Patterns» (GoF), «Refactoring» та «Clean Architecture», доповнених практиками SOLID і каталогами архітектурних стилів [5; 7; 1.14; 1.15]. Однак у більшості публікацій акцент зроблено або на загальних принципах якості коду, або на корпоративних інформаційних системах; доменно-специфічні кейси для цивільного захисту, де поєднуються геоінформаційні, оптимізаційні й симуляційні компоненти, висвітлено фрагментарно. Саме тут виникає дослідницький вакуум: як добір патернів (наприклад, Strategy, Observer, Factory, Composite) і дотримання SOLID впливають на достовірність моделювання доступності, стабільність інтеграції з алгоритмами розташування та витрати на супровід системи у довгому циклі експлуатації.

Паралельно відбувся різкий поступ у відтворюваному просторовому аналізі: відкриті дані та інструменти на кшталт OSMnx суттєво спростили побудову дорожніх графів, обчислення ізохрон і моделювання транспортної доступності для сценаріїв екстреного реагування [4]. Це створює передумови для модульних систем, де геоінформаційні обчислення, оптимізаційні моделі та шар ухвалення рішень розв'язуються окремими підсистемами з чіткими інтерфейсами. Водночас у літературі рідко порівнюють «якість архітектури» таких систем між різними підходами об'єктно-орієнтованого програмування (ООП), обмежуючись метриками продуктивності або точності просторових результатів без аналізу структурної еволюційності коду.

У сучасних роботах з управління ризиками в громадах дедалі частіше підкреслюється роль добровільних ініціатив, гнучких схем залучення та навчання, а також потреба в аналітичних платформах, що швидко пристосовуються до змін

середовища [1.1; 1.18; 1.19; 1.14]. Звідси випливає вимога до архітектури програмних рішень: підтримка сценарного аналізу, прозорої валідації даних і життєвого циклу моделей, можливість розширення джерел даних і конфігурації обмежень без «ломки» ядра. Патерни проектування тут не лише виконують «естетичну» роль гарної інженерії, а стають технічним механізмом забезпечення відтворюваності й перевіреності результатів для управлінських рішень у сфері безпеки.

Підсумовуючи огляд стану питання в науці, можна відзначити, що оптимізаційні моделі розташування рятувальних формувань достатньо формалізовані й багаторазово апробовані [2; 6; 9]. Інструментарій відкритих геоданих та мережевого аналізу дозволяє будувати реалістичні карти доступності [4], а ООП-практики з патернами задають перевірені шаблони для формування масштабованих систем [5; 7; 1.14]. Проте перехресні дослідження, які кількісно оцінюють внесок конкретних ООП-рішень і патернів в ефективність комплексних систем планування сільських рятувальних формувань (точність, стійкість до змін, вартість супроводу, час обчислень, прозорість перевірки), залишаються поза увагою науковців. Ця прогалина і визначає актуальність нашої роботи, яка стосується обґрунтування ефективності ООП-методів і патернів у задачі розробки інструментарію для підтримки управлінських рішень під час планування розташування рятувальних формувань на території сільських громад.

**Постановка завдання.** Для підвищення ефективності планування систем безпеки у сільських громадах доцільним є застосування об'єктно-орієнтованих методів програмування та патернів проектування, які дозволяють створювати гнучкі, масштабовані й відтворювані моделі підтримки управлінських рішень. Запропонований підхід передбачає поєднання класичних моделей розташування рятувальних формувань із сучасними інструментами програмної інженерії, що забезпечує можливість моделювання різних сценаріїв реагування та оптимізації просторового розташування добровільних рятувальних структур. Це дозволяє враховувати не лише транспортні та демографічні характеристики територій, а й вимоги до швидкості реагування і стійкості до змін у проєктному середовищі.

Мета дослідження полягає у визначенні ефективності методів об'єктно-орієнтованого програмування та патернів проектування під час створення моделей планування розташування

добровільних рятувальних формувань у сільських громадах. Завданням є обґрунтування того, як архітектурні рішення на основі патернів сприяють підвищенню адаптивності та масштабованості систем, а також зменшують ризики помилок під час розробки та супроводу. Запропонований підхід орієнтований на створення програмних модулів, що відображають просторову й організаційну специфіку сільських територій і дозволяють досягати таких результатів, як скорочення часу прибуття рятувальних підрозділів до місць виникнення надзвичайних ситуацій, зниження витрат на їх розташування, підвищення доступності послуг безпеки для населення та раціональне використання обмежених ресурсів громад.

**Виклад основного матеріалу.** Наше дослідження спрямоване на оцінювання ефективності методів ООП та патернів проектування під час створення моделей планування розташування добровільних рятувальних формувань для сільських громад. Основна увага приділяється визначенню того, як архітектурні рішення впливають на здатність системи бути масштабованою, адаптивною та відтворюваною.

Під ефективністю методів ООП та патернів у нашому дослідженні розуміється не лише швидкість алгоритмів, а й здатність системи до модифікації, зрозумілості її структури та зниження ризику помилок під час розробки. Для визначення ефективності вводиться інтегральний показник:

$$E = \alpha \cdot Q_r + \beta \cdot M_s + \gamma \cdot F_r + \delta \cdot C_m, \quad (1)$$

де  $Q_r$  – якість реалізації (точність і коректність роботи алгоритмів);  $M_s$  – масштабованість системи (можливість обробки більших обсягів даних без зниження продуктивності);  $F_r$  – повторне використання коду (code reusability);  $C_m$  – міра складності архітектури (complexity measure);  $\alpha, \beta, \gamma, \delta$  – відповідно вагові коефіцієнти, що визначаються експертами.

Для забезпечення об'єктивності інтегрального показника вагові коефіцієнти визначалися експертним шляхом. До експертної групи увійшли п'ять фахівців у сфері ООП та просторового моделювання рятувальних формувань, які незалежно оцінили відносну важливість кожного критерію. Узагальнення їхніх оцінок виконувалося за методом середнього зваженого, що дало змогу уникнути домінування окремих суджень та забезпечити збалансованість ваг. Такий підхід допоміг сформувати узгоджену систему пріоритетів.

Якість реалізації  $Q_r$  відображає точність і коректність роботи алгоритмів у системі. Вона формалізується через середню відносну похибку розрахунків:

$$Q_r = 1 - \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}, \quad (2)$$

де  $y_i$  – реальні значення досліджуваного показника (наприклад, час доїзду, визначений за виробничими спостереженнями або з використанням доступних сервісів);  $\hat{y}_i$  – значення досліджуваного показника, отримані програмною моделлю;  $n$  – кількість тестових прикладів.

Якщо  $Q_r$  наближається до 1, це засвідчує високу точність і надійність моделі.

Масштабованість  $M_s$  визначається здатністю системи працювати з більшими обсягами даних без суттєвої втрати продуктивності. Вона визначається за формулою

$$M_s = \frac{T(n)}{T(k \cdot n)}, \quad (3)$$

де  $T(n)$  – тривалість виконання програми за обробки  $n$  об'єктів;  $T(k \cdot n)$  – тривалість виконання при збільшенні вхідних даних у  $k$  разів.

Якщо  $M_s \approx \frac{1}{k}$ , то система масштабується лінійно. Відхилення від цього значення вказує на втрату продуктивності.

Показник повторного використання коду  $F_r$  відображає, яку частину коду може бути використано повторно в інших модулях або проектах. Цей показник визначається за формулою

$$F_r = \frac{N_{reuse}}{N_{total}} \times 100\%, \quad (4)$$

де  $N_{reuse}$  – кількість модулів (класів, методів), що були повторно використані;  $N_{total}$  – загальна кількість модулів у системі.

Високе значення  $F_r$  вказує на те, що архітектура побудована правильно, з дотриманням принципів модульності та інкапсуляції.

Показник  $C_m$  використовується для оцінювання складності архітектури, що відображає цикломатичну складність (Cyclomatic Complexity):

$$C_m = R - N + 2P, \quad (5)$$

де  $R$  – кількість ребер у графі керування програмою;  $N$  – кількість вершин у графі;  $P$  – кількість незалежних компонентів (наприклад, класів або підпрограм).

Чим вище значення  $C_m$ , тим складніше супроводжувати систему. Оптимальним вважається рівень  $C_m \leq 10$  для окремого модуля.

Таким чином, кожна складова ефективності методів ООП та патернів проектування має свої кількісні значення, які визначаються за формулами (2), (3), (4) і (5). Об'єднання цих показників в інтегральну формулу (1) дозволяє комплексно оцінити ефективність методів ООП та патернів під час створення моделей планування розташування добровільних рятувальних формувань для сільських громад. При цьому ефективність розглядається як багатокритеріальний показник, що поєднує технічні та організаційні чинники.

Пропонується програмну модель будувати на принципах SOLID, що забезпечує зрозумілість і стійкість коду. Цей принцип визначає базові правила ООП, які допомагають створювати зрозумілий, гнучкий і легко підтримуваний код. Його застосування забезпечує стійкість архітектури до змін та зменшує ризик виникнення помилок при розширенні функціоналу. Зокрема, до них належать:

- Single Responsibility Principle – гарантує, що кожен клас відповідає лише за одну функцію;
- Open / Closed Principle – забезпечує розширюваність системи без зміни вже написаного коду;
- Dependency Inversion Principle – дає можливість змінювати алгоритми через абстракції.

Для ілюстрації використання цих принципів побудовано UML-діаграму (рис. 1), яка описує архітектуру системи планування розташування добровільних рятувальних формувань для сільських громад.

Система планування розташування добровільних рятувальних формувань для сільських громад побудована з урахуванням принципів модульності та повторного використання коду. На UML-діаграмі (див. рис. 1) представлені класи, що відповідають за опис громади, рятувальних формувань, дорожньої інфраструктури, а також сервіси для обробки даних, оптимізації та візуалізації.

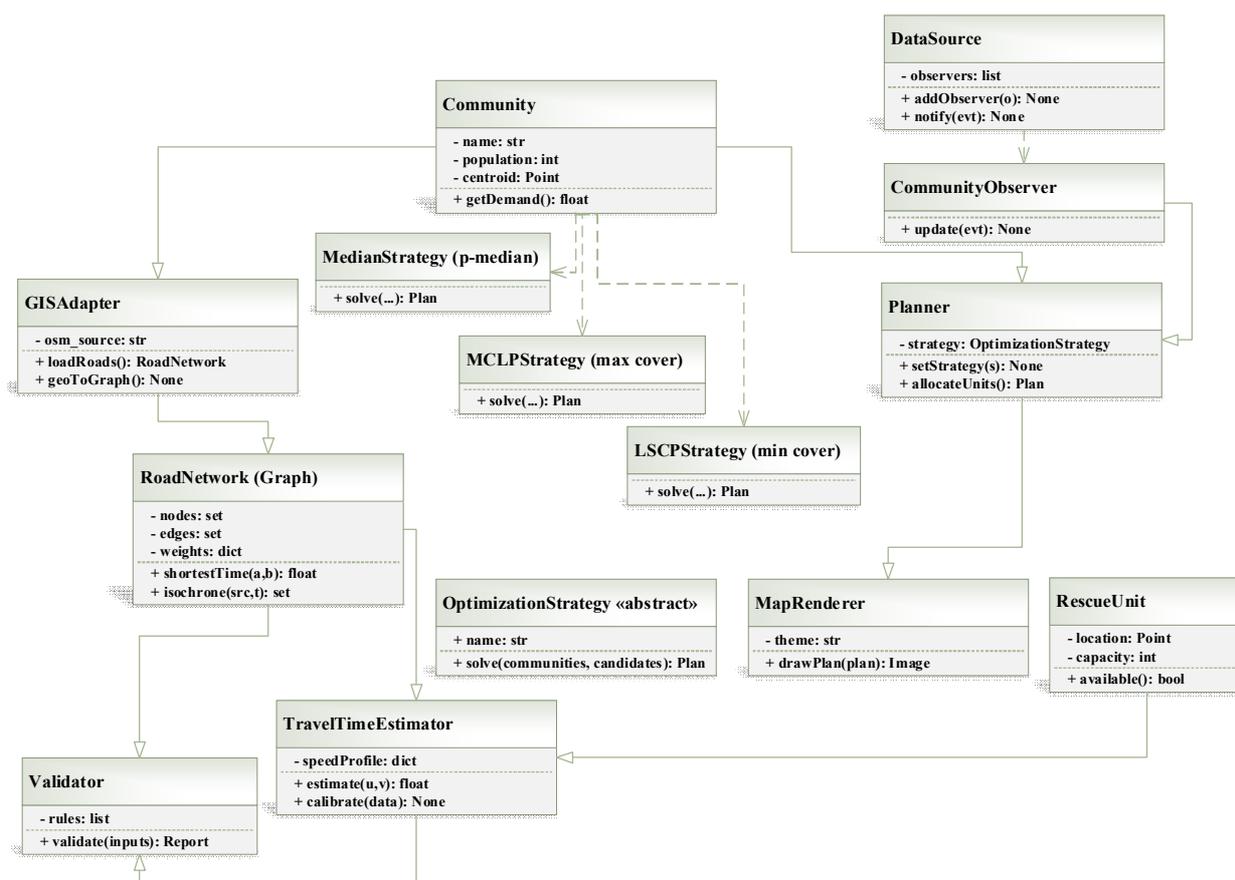


Рис. 1. Узагальнена UML-діаграма системи планування розташування добровільних рятувальних формувань для сільських громад

Fig. 1. Generalized UML diagram of the system for planning the location of voluntary rescue teams for rural communities

Клас `Community` моделює сільську громаду, враховуючи її назву, кількість населення та географічний центр. Він забезпечує розрахунок потреби в рятувальних ресурсах. Клас `RescueUnit` представляє окреме рятувальне формування з вказанням місця розташування та конфігурації, а також дозволяє перевірити його доступність для реагування на надзвичайні ситуації. Разом ці класи формують основу вхідних даних для системи.

Для моделювання транспортної мережі використовується клас `RoadNetwork`, який описує вузли, ребра та ваги графа. Саме на його основі виконуються розрахунки часу переміщення підрозділів. Завантаження й перетворення геоінформаційних даних здійснюється через клас `GISAdapter`, який отримує дані з відкритих картографічних ресурсів і формує з них дорожній граф. Для підвищення точності оцінки використовується клас `TravelTimeEstimator`, який, спираючись на профілі швидкості, оцінює час доїзду рятувальних формувань до громад. Коректність даних і побудованих моделей перевіряє клас `Validator`, що знижує ризик отримання помилкових результатів.

Ядром системи виступає клас `Planner`, який реалізує патерн `Strategy`. Він дозволяє динамічно підключати різні алгоритми оптимізації залежно від завдань – мінімізація середнього часу доїзду (`MedianStrategy`), максимізація охоплення території (`MCLPStrategy`) або мінімізація кількості формувань для забезпечення повного покриття (`LSCPStrategy`) території громади. Базовий абстрактний клас `OptimizationStrategy` визначає єдиний інтерфейс для всіх стратегій. Завдяки цьому архітектура легко розширюється новими методами оптимізації без зміни основного планувальника.

Важливим елементом є клас `MapRenderer`, який відповідає за візуалізацію результатів планування на карті. Це забезпечує зрозуміле представлення інформації для користувачів і ухвалення управлінських рішень.

Система також реалізує патерн `Observer`, що дає змогу автоматично реагувати на зміни вхідних даних. Клас `DataSource` виступає генератором подій, а `CommunityObserver` отримує оновлення й передає їх до `Planner` для повторного виконання оптимізації. Такий підхід забезпечує адаптивність та актуальність рішень навіть за динамічних умов у громадах.

Представлена діаграма (див. рис. 1) відображає комплексну архітектуру, яка охоплює всі етапи, починаючи від збору та перевірки даних до вибору оптимального сценарію розташування рятувальних формувань та візуалізації результатів.

Використання патернів `Strategy` та `Observer` забезпечує гнучкість, стійкість і можливість масштабування системи.

На рис. 2 наведено фрагмент програмного коду, реалізованого мовою Python, що ілюструє використання патерну `Strategy` в системі планування розташування добровільних рятувальних формувань на території сільських громад.

```
from abc import ABC, abstractmethod
from typing import List

# Абстрактна стратегія оптимізації
class OptimizationStrategy(ABC):
    @abstractmethod
    def solve(self, communities: List[str], units: int) -> dict:
        pass

# Стратегія p-median
class MedianStrategy(OptimizationStrategy):
    def solve(self, communities, units):
        return {"method": "p-median", "units": units, "result": "Min avg time"}

# Стратегія максимального покриття
class MCLPStrategy(OptimizationStrategy):
    def solve(self, communities, units):
        return {"method": "MCLP", "units": units, "result": "Max coverage"}

# Планувальник, що реалізує патерн Strategy
class Planner:
    def __init__(self, strategy: OptimizationStrategy):
        self.strategy = strategy

    def set_strategy(self, strategy: OptimizationStrategy):
        self.strategy = strategy

    def allocate_units(self, communities, units):
        return self.strategy.solve(communities, units)

# Демонстрація використання
communities = ["Village A", "Village B", "Village C"]
planner = Planner(MedianStrategy())
print(planner.allocate_units(communities, 3))

planner.set_strategy(MCLPStrategy())
print(planner.allocate_units(communities, 3))
```

**Рис. 2.** Фрагмент програмного коду реалізації патерну `Strategy` в системі планування розташування добровільних рятувальних формувань для сільських громад

**Fig. 2.** Fragment of the program code implementing the `Strategy` pattern in the system for planning the location of voluntary rescue teams for rural communities

У представленому фрагменті коду (див. рис. 2) показано абстрактний клас стратегії оптимізації та його конкретні реалізації. Зокрема, метод `p-median` для мінімізації середнього часу доїзду до місць виникнення надзвичайних ситуацій та метод `MCLP` для максимального охоплення території. Клас `Planner` виконує роль керуючого модуля, який дозволяє динамічно змінювати обрану стратегію та формувати план розташування підрозділів. Подібний підхід забезпечує гнучкість архітектури й адаптивність системи до різних сценаріїв функціонування рятувальних формувань.

На рис. 3 наведено фрагмент програмного коду, який демонструє реалізацію патерну `Observer` у системі планування розташування добровільних рятувальних формувань для сільських громад.

У представленому фрагменті коду (див. рис. 3) клас DataSource виступає генератором подій, а клас CommunityObserver реагує на зміни даних і передає їх до планувальника для повторного розрахунку оптимального розташування. Використання цього патерну забезпечує динамічне оновлення рішень системи у разі появи нових даних або зміни умов, що є особливо важливим для підвищення оперативності й актуальності управлінських рішень у громадах.

Поєднання методів ОПП та запропонованих патернів створює цілісну методологію, в якій математичні моделі оптимізації інтегруються з архітектурними принципами програмної інженерії. Методи ООП забезпечують модульність, інкапсуляцію та спадкування, що дозволяє будувати систему у вигляді взаємопов'язаних компонентів. Патерни, своєю чергою, надають цим компонентам гнучкості та динамічності. Зокрема, патерн Strategy дозволяє легко змінювати алгоритми планування залежно від умов, а патерн Observer забезпечує автоматичне оновлення даних та реакцію системи на зовнішні події.

```

from abc import ABC, abstractmethod

# Абстрактний спостерігач
class Observer(ABC):
    @abstractmethod
    def update(self, data):
        pass

# Конкретний спостерігач (громада)
class CommunityObserver(Observer):
    def update(self, data):
        print(f"CommunityObserver отримав нові дані: {data}")

# Джерело даних
class DataSource:
    def __init__(self):
        self.observers = []

    def add_observer(self, observer: Observer):
        self.observers.append(observer)

    def notify(self, data):
        for obs in self.observers:
            obs.update(data)

# Демонстрація роботи патерну Observer
source = DataSource()
observer1 = CommunityObserver()
observer2 = CommunityObserver()

source.add_observer(observer1)
source.add_observer(observer2)

# При надходженні нових даних усі спостерігачі автоматично оновлюються
source.notify("Оновлені параметри громади")

```

**Рис. 3.** Фрагмент програмного коду реалізації патерну

Observer для автоматичного оновлення даних у системі планування розташування добровільних рятувальних формувань для сільських громад

**Fig. 3.** Fragment of the program code implementing the Observer pattern for automatic data updating in the system for planning the location of voluntary rescue teams for rural communities

Синергія між ООП-методами та патернами проявляється у здатності системи одночасно бути структурованою та адаптивною. Формальна архітектура на основі класів та інтерфейсів створює передумови для повторного використання коду, тоді як застосування патернів мінімізує залежності між модулями і знижує складність масштабування. У результаті розробник отримує не лише інструмент для моделювання розташування рятувальних формувань, а й гнучку платформу, яку можна розширювати новими методами оптимізації та сценаріями використання.

Запропоноване поєднання є особливо важливим у системі планування розташування добровільних рятувальних формувань для сільських громад. У ній слід забезпечити швидку адаптацію до змін проектного середовища. Завдяки ООП-методам система зберігає логічну цілісність і зрозумілу структуру, а завдяки патернам – здатність до оперативної перебудови та інтеграції нових алгоритмів. Таким чином, методологія демонструє, що саме інтеграція цих двох підходів формує основу ефективних рішень для підвищення ефективності планування розташування добровільних рятувальних формувань для сільських громад.

Оцінювання ефективності методів ООП та патернів проектування під час створення моделей планування розташування добровільних рятувальних формувань проводилося для умов Шептицької міської територіальної громади Львівської області. Було змодельовано умови, наближені до реальних:

- територія охоплювала 15 населених пунктів з різною кількістю населення;
- у дорожній мережі враховувалася доступність доріг місцевого та регіонального значення, середня швидкість руху автотранспорту становила 60 км/год;
- оцінювався час доїзду рятувальних підрозділів до населених пунктів, а також можливість масштабування системи при збільшенні кількості об'єктів;
- програмна модель будувалася трьома різними підходами – процедурним, класичним ООП без патернів і ООП з використанням патернів Strategy та Observer.

Оцінювання здійснювалося за такими показниками: якість реалізації  $Q_r$ , масштабованість  $M_s$ , повторне використання коду  $F_r$  та складність архітектури  $C_m$ . З огляду на завдання щодо планування розташування добровільних рятувальних формувань, де особливо важливими є якість реалізації та масштабованість, вагові коефіцієнти були визначені експертним шляхом та прийнято  $\alpha = 0.35$ ,  $\beta = 0.30$ ,  $\gamma = 0.20$ ,  $\delta = 0.15$ . Ці значен-

ня забезпечують пріоритетність коректності системи та її здатності до зростання, водночас враховуючи важливість повторного використання модулів і зменшення складності архітектури.

Результати визначення показників інтегральної ефективності методів ОПП та патернів у системі планування розташування добровільних рятувальних формувань для сільських громад представлено у таблиці.

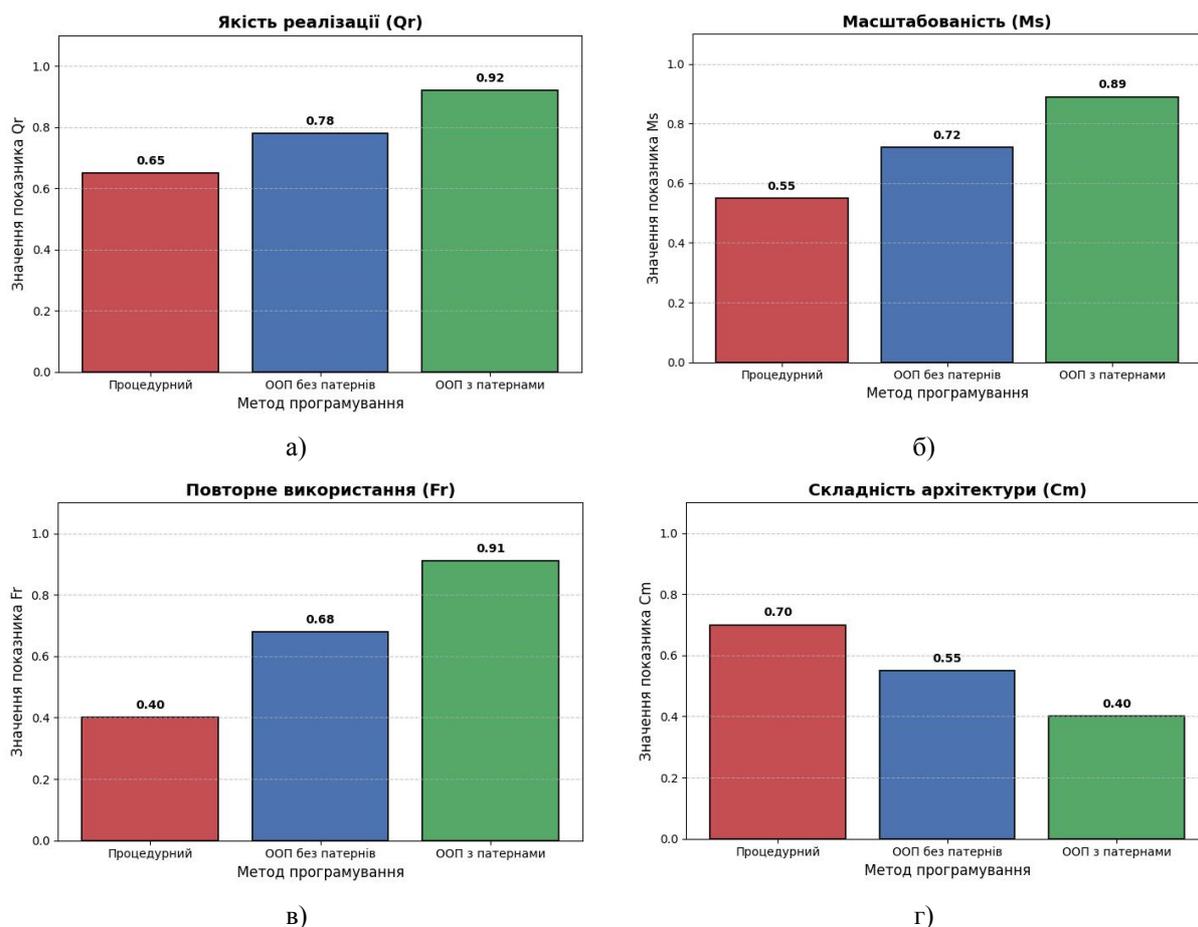
На основі отриманих результатів та використання формули (1) встановлено, що інтегральна ефективність процедурного підходу становить лише

0.58. Це свідчить про обмежені можливості використання цього методу в реальних умовах. Для класичного ООП значення ефективності зросло до 0.73 завдяки покращанню масштабованості та повторного використання коду. Найвищі показники зафіксовано для варіанта ООП із патернами, де інтегральна ефективність досягла 0.87. Це підтверджує гіпотезу, що саме поєднання ООП із патернами проектування забезпечує найкращі результати в системі планування розташування добровільних рятувальних формувань для сільських громад.

**Таблиця.** Результати визначення показників інтегральної ефективності методів ОПП та патернів у системі планування розташування добровільних рятувальних формувань для сільських громад

**Table.** Results of determining the indicators of the integral effectiveness of object-oriented programming methods and patterns in the system of planning the location of voluntary rescue teams for rural communities

Метод	$Q_r$	$M_s$	$F_r$	$C_m$	$E$
Процедурний підхід	0.65	0.55	0.40	0.70	0.58
ООП без патернів	0.78	0.72	0.68	0.55	0.73
ООП із патернами	0.92	0.89	0.91	0.40	0.87



**Рис. 4.** Діаграми показників якості реалізації  $Q_r$  (а), масштабованості  $M_s$  (б), повторного використання коду  $F_r$  (в) та складності архітектури  $C_m$  (г) за різних використовуваних методів програмування

**Fig. 4.** Diagrams of indicators of implementation quality  $Q_r$  (a), scalability  $M_s$  (b), code reuse  $F_r$  (c), and architectural complexity  $C_m$  (d) for different programming methods used

Ми порівняли показник якості реалізації  $Q$ , за різних використовуваних методів програмування (див. рис. 4).

Представлені на рис. 4 діаграми підтверджують висновки з таблиці. Найбільший прогрес досягається при переході від процедурного підходу до об'єктно-орієнтованого, однак саме застосування патернів дає найбільший ефект. Якість реалізації зростає від 0.65 до 0.92, а повторне використання коду – більш ніж удвічі (від 0.40 до 0.91). Масштабованість системи у варіанті з патернами становить 0.89, що підтверджує можливість обробки більших обсягів даних без втрати продуктивності. Водночас складність архітектури зменшилася до 0.40, що робить систему зрозумілою та легкою у супроводі.

На підставі проведених досліджень встановлено, що запропонований підхід, який ґрунтується на поєднанні методів ООП і патернів проектування, є найефективнішим для задачі планування розташування добровільних рятувальних формувань для сільських громад. Він забезпечує високий рівень точності та коректності, дозволяє повторно використовувати модулі й легко інтегрувати нові алгоритми, знижуючи складність архітектури. Отримані результати доводять доцільність впровадження такого підходу у практику розробки інформаційних систем для цивільного захисту, де особливо важливими є адаптивність і швидке реагування на зміни.

**Висновки.** 1. Виконаний аналіз стану питання в теорії та практиці свідчить, що процедурні підходи до розробки програмних систем для планування розташування добровільних рятувальних формувань для сільських громад мають низьку гнучкість, слабо масштабовані та характеризуються високою складністю супроводу. Класичне об'єктно-орієнтоване програмування частково зменшує ці недоліки. Однак воно не забезпечує належної адаптивності за врахування зміни зовнішніх умов чи алгоритмів. Пропонується використовувати підхід, який поєднує принципи об'єктно-орієнтованого програмування з патернами проектування.

2. Запропоновано підхід до оцінювання ефективності методів об'єктно-орієнтованого програмування і патернів для планування сільських рятувальних формувань, який поєднує використання класичних моделей просторового розташування рятувальних формувань із застосуванням принципів SOLID і патернів проектування Strategy та Observer. Такий підхід дозволяє створити

архітектуру, що легко масштабується, має нижчу складність та забезпечує повторне використання компонентів.

3. На основі проведених досліджень із розробленими програмними модулями на мові Python 3.12 здійснено оцінювання ефективності методів об'єктно-орієнтованого програмування та патернів проектування під час створення моделей планування розташування добровільних рятувальних формувань, яке проводилося для умов Шептицької міської територіальної громади Львівської області. Встановлено, що процедурний підхід забезпечує найнижчі показники. Зокрема, показник якості реалізації становить 0.65, масштабованість – 0.55, повторного використання компонентів – 0.40. Водночас складність залишалася високою – 0.70. Використання класичного об'єктно-орієнтованого програмування покращило показники якості до 0.78, масштабованості – до 0.72 та повторного використання – до 0.68. При цьому складність становила 0.55. Найкращі результати отримано під час застосування об'єктно-орієнтованого програмування із патернами. Якість реалізації зросла до 0.92, масштабованість – до 0.89, повторне використання – до 0.91, а складність знизилася до 0.40. Це підтверджує переваги інтеграції патернів проектування у системи планування розташування добровільних рятувальних формувань для сільських громад.

4. Надалі слід розширити набір алгоритмів і стратегій оптимізації, залучити стохастичні та робастні методи, інтегрувати динамічні просторові дані та чинники мобільності населення. Окремо слід провести верифікацію на реальних прикладах різних сільських громад та оцінити вплив інших патернів проектування на показники ефективності програмування.

#### Бібліографічний список

1. Оптимізація безпекової інфраструктури у програмах післявоєнного відновлення з використанням сучасних геоінформаційних систем / А. М. Тригуба, Р. Т. Ратушний, А. Р. Ратушний, Л. С. Коваль, А. І. Івануса. *Вісник Львівського державного університету безпеки життєдіяльності*. 2025. № 31. С. 196–212. doi: <https://doi.org/10.32447/20784643.31.2025.20>.
2. Adasme P., Viveros A., Dehghan Firoozabadi A. Quadratic p-Median Problem: A Bender's Decomposition and a Meta-Heuristic Local-Based Approach. *Symmetry*. 2024. Vol. 16, No. 9. P. 1114. doi: <https://doi.org/10.3390/sym16091114>.
3. Alexander D. Disaster and Emergency Planning for Preparedness, Response, and Recovery. *Oxford Research Encyclopedia of Natural Hazard Science*. 2015. doi: <https://doi.org/10.1093/acrefore/9780199389407.013.12>.

4. Algorithm for the routes formation of food raw materials procurement on the community territory taking into account the production conditions during emergency situations / A. M. Tryhuba, N. Ya. Koval, A. R. Ratushnyi, I. L. Tryhuba, V. V. Shevchuk. *Applied Aspects of Information Technology*. Odessa: Nauka i Tekhnika, 2023. Vol. 6, No. 1. P. 60–73. doi: <https://doi.org/10.15276/aait.06.2023.5>.
5. Almadi S. H. S., Hooshyar D., Ahmad R. B. Bad smells of Gang of Four design patterns: a decade systematic literature review. *Sustainability*. 2021. Vol. 13, No. 18. P. 10256. doi: <https://doi.org/10.3390/su131810256>.
6. A Neural Network Model for Predicting the Duration of Emergency Response Projects / A. Ratushnyi, V. Ptashnyk, L. Koval, P. Lub, A. Tatomyr. *IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)*. 2023. 195025. P. 1–4. doi: <https://doi.org/10.1109/csit61576.2023.10324279>.
7. Bafandeh Mayvan B., Rasoolzadegan A., Ghavidel Yazdi Z. The state of the art on design patterns: A systematic mapping of the literature. *Journal of Systems and Software*. 2016. Vol. 125. P. 93–118. doi: <https://doi.org/10.1016/j.jss.2016.11.030>.
8. Continuous Maximum Coverage Location Problem with Arbitrary Shape of Service Areas and Regional Demand / S. Yakovlev et al. *Symmetry*. 2025. Vol. 17. P. 676.
9. Delineations for police patrolling on street network segments with p-median location models / C. Lee, H. Kim, Y. Chun, D. A. Griffith. *ISPRS International Journal of Geo-Information*. 2024. Vol. 13, No. 11. P. 410. doi: <https://doi.org/10.3390/ijgi13110410>.
10. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. Addison-Wesley, 1995.
11. Intelligent tools for selecting configurations of renewable energy equipment for livestock farms. / T. Hutsol, K. Mudryk, S. Francik, R. Oliinyk. *Energies*. 2023. Vol. 16(22). P. 7605. doi: <https://doi.org/10.3390/en16227605>.
12. Kondysiuk I., Tryhuba I., Boiarchuk O. Intellectual information system for formation of portfolio projects of motor transport enterprises. *CEUR Workshop Proceedings*. Vol. 3109. P. 44–52. URL: <https://ceur-ws.org/Vol-3109/paper7.pdf> (Accessed February 09, 2025).
13. Lamontagne S., Carvalho M., Atallah R. Accelerated Benders decomposition and local branching for dynamic maximum covering location problems. *Computers & Operations Research*. 2024. Vol. 167. 106673. doi: <https://doi.org/10.1016/j.cor.2024.106673>.
14. Murray A. T. Beyond location modeling and GIS: Integration and bridging. *Computers & Operations Research*. 2025. Vol. 180. P. 107073. doi: <https://doi.org/10.1016/j.cor.2025.107073>.
15. On microservice analysis and architecture evolution: a systematic mapping study / V. Bushong et al. *Applied Sciences*. 2021. Vol. 11, No. 17. P. 7856. doi: <https://doi.org/10.3390/app11177856>.
16. Optimal charging facility location and capacity for electric vehicles considering route choice and charging time equilibrium / R. Chen, X. Qian, L. Miao, S. V. Ukkusuri. *Computers & Operations Research*. 2020. Vol. 113. P. 104776. doi: <https://doi.org/10.1016/j.cor.2019.104776>.
17. Substantiating the effectiveness of projects for the construction of dual systems of fire suppression / R. Ratushnyi, P. Khmel, E. Martyn, O. Prydatko. *Eastern-European Journal of Enterprise Technologies*. 2019. Vol. 4(3-100). P. 46–53.
18. The Value Formation Model of the Project's Implementation for the Territorial Rescue Structures Creation to a Consequences Elimination of a Military Emergency Situations / A. Tryhuba, A. Ratushnyi, P. Lub, M. Rudynets, O. Visyn. *Proceedings of the 4th International Workshop IT Project Management (ITPM 2023). Conference*. Warsaw, 2023. P. 59–70. URL: <https://ceur-ws.org/Vol-3453/paper6.pdf> (Accessed February 09, 2025).
19. Twigg J., Mosel I. Emergent groups and spontaneous volunteers in urban disaster response. *Environment and Urbanization*. 2017. Vol. 29, No. 2. P. 443–458. doi: <https://doi.org/10.1177/0956247817721413>.
20. Value-based hybrid management of community safety projects: intelligent models for optimal location of volunteer rescue formations / A. Tryhuba, O. Malanchuk, A. Ratushnyi, L. Koval, O. Andrushkiv. *Sustainability*. 2024. Vol. 16(12). P. 4821. doi: <https://doi.org/10.3390/su16124821>.

Стаття надійшла 12.02.2025