

УДК 004.434

ЗАСТОСУВАННЯ ПРИНЦИПІВ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ В C# ПРИ РОЗРОБЦІ СУЧАСНИХ ПРОГРАМ

Мурзабулатова М.С., Дейнеко Ж.В.

e-mail: mariia.murzabulatova@nure.ua, zhanna.deineko@nure.ua

Харківський національний університет радіоелектроніки, каф. МСТ
м. Харків, Україна

The abstract examines the practical aspects and benefits of applying the Object-Oriented Programming paradigm in C# software development, with particular attention to encapsulation, inheritance, polymorphism, and abstraction as the conceptual basis for building flexible, scalable, and reliable software systems. The study presents applied examples from enterprise solutions and user interface applications on the .NET platform, emphasizing the role of OOP in supporting a long software lifecycle, facilitating modernization, and substantiating the relevance of an architectural approach to code design in modern software engineering.

У сучасних умовах розробка програмного забезпечення орієнтується не лише на створення функціонального продукту, а й на забезпечення його гнучкості, масштабованості, надійності та зручності супроводу. За таких умов особливого значення набуває об'єктно-орієнтоване програмування як підхід, що забезпечує моделювання предметної області через систему взаємопов'язаних об'єктів. Однією з мов, у якій принципи об'єктно-орієнтованого програмування реалізовано найбільш послідовно, є C# [1]. Завдяки платформі .NET вона широко використовується для створення настільних, веборієнтованих і корпоративних застосунків, що зумовлює актуальність аналізу її можливостей у контексті реалізації базових принципів ООП [1-3].

Метою роботи є аналіз особливостей реалізації фундаментальних принципів об'єктно-орієнтованого програмування засобами мови C# та обґрунтування їх значення для забезпечення архітектурної цілісності, масштабованості й надійності сучасних програмних продуктів.

Об'єктно-орієнтований підхід передбачає побудову програми як сукупності класів і об'єктів, що дає змогу логічно відобразити структуру системи, процеси та зв'язки між її компонентами [2]. За такого підходу програмний код формується як цілісна система взаємопов'язаних елементів, а не як набір ізольованих фрагментів. Це створює підґрунтя для більш упорядкованого проєктування, спрощує розробку складних програмних рішень і підвищує якість програмного продукту.

Концептуальне представлення взаємозв'язку між середовищем C#, фундаментальними принципами ООП, сферами їх практичної реалізації та ключовими характеристиками програмного продукту подано на рис. 1.

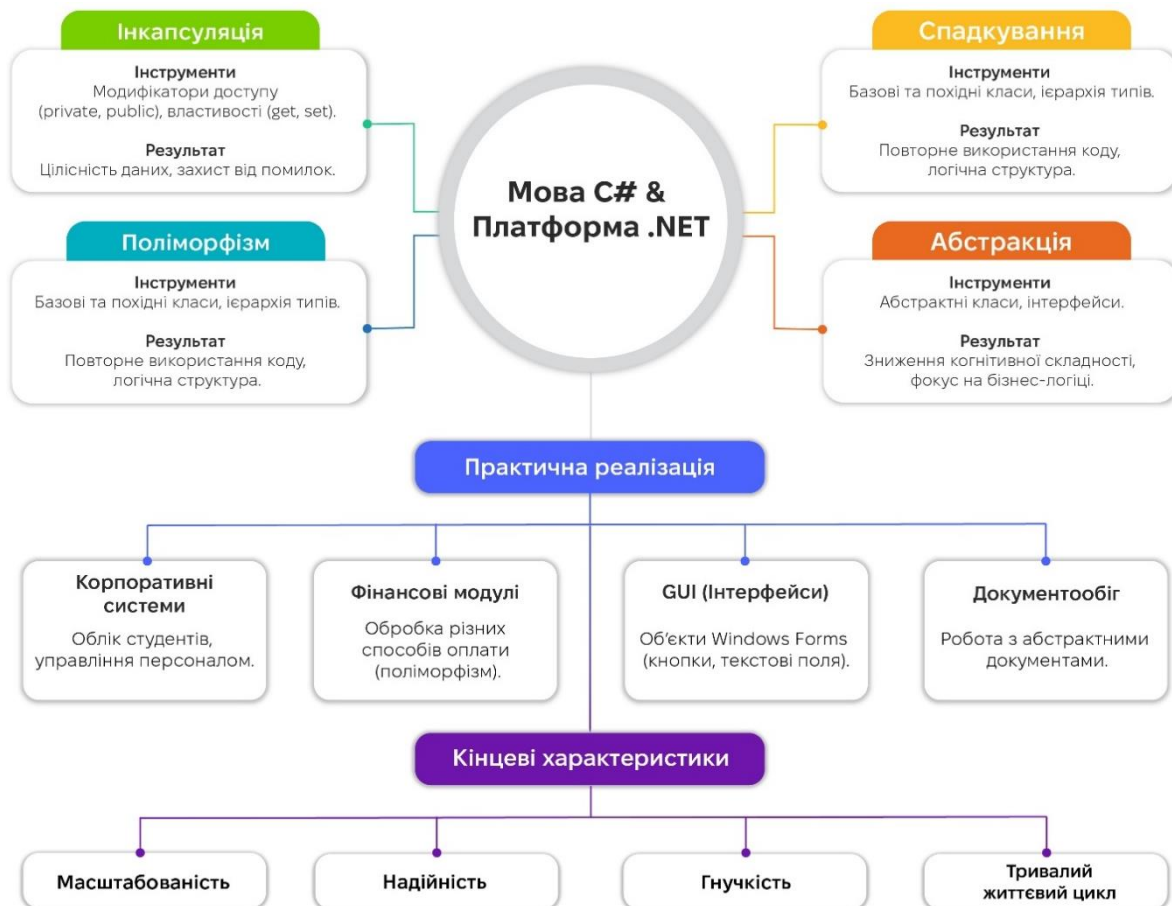


Рисунок 1 – Екосистема об'єктно-орієнтованого програмування у розробці програмного коду на C#

Одним із ключових принципів ООП є інкапсуляція, сутність якої полягає в об'єднанні даних і засобів їх обробки в межах одного класу, а також у приховуванні внутрішньої реалізації [2, 3]. У мові C# цей принцип реалізується за допомогою модифікаторів доступу, властивостей і методів, що забезпечує цілісність даних, контроль доступу до них і зниження ризику помилок під час функціонування програми.

Важливе значення має спадкування, яке дає змогу створювати нові класи на основі вже існуючих. У середовищі C# цей механізм забезпечує повторне використання коду та сприяє побудові логічної ієрархії типів. Такий підхід є доцільним у випадках, коли окремі компоненти системи мають спільні характеристики, але відрізняються за функціональними можливостями або особливостями поведінки.

Не менш важливим принципом є поліморфізм, який забезпечує можливість різної реалізації однієї й тієї самої операції залежно від типу об'єкта. У C# цей механізм дає змогу організувати взаємодію з різними об'єктами в межах єдиного підходу, що підвищує гнучкість програмної системи та спрощує модифікацію її окремих елементів без порушення загальної структури коду.

Суттєву роль у побудові програмних систем відіграє абстракція, яка дає змогу зосередити увагу на ключових функціях об'єкта без надмірної деталізації внутрішніх механізмів реалізації. Практичне значення цього принципу полягає у спрощенні роботи зі складними компонентами системи та зменшенні когнітивного навантаження під час проектування і супроводу програмного продукту.

Практична значущість зазначених принципів простежується у розробці корпоративних інформаційних систем, застосунків для управління персоналом, фінансових модулів, систем документообігу та програм із графічним інтерфейсом. Використання ООП у С# набуває особливого значення під час створення програмних продуктів із тривалим життєвим циклом, оскільки такий підхід спрощує налагодження, тестування, модернізацію та подальше розширення функціональних можливостей програмної системи. Разом із цим результативність застосування об'єктно-орієнтованого підходу визначається не лише технічними можливостями мови, а й якістю архітектурного проектування, оскільки формальне використання класів без чіткого розподілу відповідальності між ними не забезпечує належного рівня якості програмного продукту.

Висновки. Проведений аналіз дає підстави стверджувати, що реалізація фундаментальних принципів об'єктно-орієнтованого програмування в мові С# є важливою умовою побудови сучасних програмних систем, орієнтованих на надійність, гнучкість і масштабованість. Інкапсуляція, спадкування, поліморфізм та абстракція забезпечують структурованість програмного коду, підвищують ефективність його супроводу та створюють підґрунтя для подальшої модернізації програмного продукту. Це дозволяє розглядати мову С# як доцільний інструмент реалізації сучасних підходів до проектування програмного забезпечення, розрахованого на тривалий життєвий цикл і високу архітектурну якість.

Список використаних джерел:

1. Heričko, T., & Šumak, B. (2023). Exploring Maintainability Index Variants for Software Maintainability Measurement in Object-Oriented Systems. *Applied Sciences*, 13(5), Art. 2972. DOI: <https://doi.org/10.3390/app13052972>.
2. Щербаков, О.В., Парфьонов, Ю.Е., & Федорченко, В.М. (2019). *Основи об'єктно-орієнтованого програмування: навчальний посібник*. Харків: ХНЕУ ім. С. Кузнеця.
3. Коноваленко, І.В., Марущак, П.О., & Савків, В.Б. (2017). *Програмування мовою С# 7.0: навчальний посібник*. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя. https://elartu.tntu.edu.ua/bitstream/lib/22436/1/Konovalenko_I_Programuvanny_a_C%23_2017.pdf.