

Рішення
разової спеціалізованої вченої ради
про присудження ступеня доктора філософії

Здобувач ступеня доктора філософії: Форкерт Павло Павлович, 1988 року народження, громадянин України, освіта вища: закінчив у 2010 році Дніпропетровський національний університет імені Олеся Гончара за спеціальністю «Програмне забезпечення автоматизованих систем», навчається в аспірантурі Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України з 2022 р., виконав акредитовану освітньо-наукову програму «Інженерія програмного забезпечення».

Разова спеціалізована вчена рада, утворена наказом Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України, м. Дніпро, від «06» липня 2026 року № 248, у складі:

голови разової спеціалізованої вченої ради

- Олега БАЙБУЗА, доктора технічних наук, професора, завідувача кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України;

рецензентів:

- Світлани АНТОНЕНКО, кандидатки технічних наук, доцентки, доцентки кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України;
- Лілії БОЖУХИ, кандидатки фізико-математичних наук, доцентки, доцентки кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України;

офіційних опонентів:

- Андрія ГУБСЬКОГО, кандидата технічних наук, доцента кафедри інформатики та програмної інженерії факультету інформатики і обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Міністерства освіти і науки України;
- Максима ЄВЛАНОВА, доктора технічних наук, професора, професора кафедри інформаційних управляючих систем факультету комп'ютерних наук Харківського національного університету радіоелектроніки Міністерства освіти і науки України;

на засіданні «25» серпня 2026 року прийняли рішення про присудження ступеня доктора філософії з галузі знань 12 Інформаційні технології Форкерту

Павлу Павловичу на підставі публічного захисту дисертації «Дослідження використання рантайму Go як платформи для побудови нових мов програмування» за спеціальністю 121 Інженерія програмного забезпечення.

Дисертацію виконано у Дніпровському національному університеті імені Олеся Гончара Міністерства освіти і науки України, м. Дніпро.

Науковий керівник: ІВАНЧЕНКО Марина Геннадіївна, кандидатка технічних наук, доцентка, доцентка кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України.

Дисертацію подано у вигляді спеціально підготовленого рукопису із дотримання вимог пункту 6 Порядку присудження ступеня доктора філософії та скасування рішення разової спеціалізованої вченої ради закладу вищої освіти, наукової установи про присудження ступеня доктора філософії, затвердженого постановою Кабінету Міністрів України від 12 січня 2022 року № 44 (зі змінами).

У дисертації вперше системно обґрунтовано доцільність розгляду рантайму Go як хост-платформи для побудови нових мов програмування, тобто як цілі транспіляції для мов із високою інтероперабельністю з екосистемою Go. Також вперше в межах єдиного транспіляційного підходу показано можливість реалізації комплексу мовних розширень поверх Go-рантайму без модифікації офіційного компілятора Go, зокрема повнофункціональних `enum`-типів, зіставлення із взірцем, іменованих аргументів, значень параметрів за замовчуванням, методів розширення, узагальнених методів, коротких анонімних функцій та універсального синтаксису виклику функцій. Удосконалено підхід до оцінювання придатності Go як цілі транспіляції шляхом комплексного врахування виконувальних властивостей, виразних можливостей хост-мови, інтероперабельності, інструментальної підтримки транспіляції та вартості реалізації мовних конструкцій. Удосконалено також архітектурний підхід до побудови мовних розширень у формі надмножини Go з поетапною трансформацією у звичайний Go-код та збереженням двосторонньої інтероперабельності з екосистемою Go, який поєднує нормалізацію конструкцій, частковий семантичний аналіз і перетворення у вузли `go/ast`.

Практичне значення роботи полягає в тому, що запропонований підхід дає змогу суттєво зменшити витрати на створення нових мов програмування, діалектів і предметно-орієнтованих мов за рахунок повторного використання компілятора, рантайму, системи збірки та бібліотек Go. Для демонстрації цього в межах роботи було розроблено експериментальний транспілятор діалекту GoNext на основі офіційних інструментів аналізу та генерації коду Go, що підтверджує практичну здійсненність запропонованих рішень. Результати роботи можуть бути використані для проектування нових серверних, інтеграційних, інфраструктурних і конфігураційних мов, для створення навчальних і дослідницьких транспіляторів, а також для

еволюційного розширення мови Go без втрати сумісності з уже наявним кодом і бібліотеками Go. Результати дисертаційної роботи впроваджено в освітній процес Дніпровського національного університету імені Олеся Гончара під час викладання освітньої компоненти «Мовні технології».

Дисертація виконана державною мовою із дотриманням вимог до оформлення дисертації, встановлених МОН України. Обсяг основного тексту дисертації відповідає нормам, встановленим освітньо-науковою програмою «Інженерія програмного забезпечення» за спеціальністю 121 Інженерія програмного забезпечення Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України.

Здобувач має 11 наукових публікацій за темою дисертації, з яких 3 статті у наукових фахових виданнях України категорії «Б» та 8 тез доповідей у збірниках матеріалів наукових конференцій, які відповідають вимогам пунктів 8, 9 Порядку присудження ступеня доктора філософії та скасування рішення разової спеціалізованої вченої ради закладу вищої освіти, наукової установи про присудження ступеня доктора філософії, зокрема:

1. Forkert P. P., Sydorova M. G. Integrating full-featured enums into Go programming language. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 27, Дніпро, 2023, с. 3-16. DOI: <http://dx.doi.org/10.15421/432301> URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/226> (фахове видання категорії «Б»).

2. Forkert P. P., Ivanchenko M. G. Implementing named arguments in Go programming language dialect. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 29, Дніпро, 2025, с. 3-11. DOI: <https://dx.doi.org/10.15421/432501> URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/290> (фахове видання категорії «Б»).

3. Forkert P. P., Ivanchenko M. G. Implementing extension methods and generic methods in Go programming language dialect. *Системні технології*, т. 163, Дніпро, 2026, с. 111-121. DOI: <https://doi.org/10.34185/1562-9945-2-163-2026-10> URL: <https://journals.nmetau.edu.ua/index.php/st/en/article/view/2300> (фахове видання категорії «Б»).

У дискусії взяли участь голова і члени спеціалізованої ради та присутні на захисті фахівці:

Голова спеціалізованої вченої ради – Байбуз Олег Григорович, доктор технічних наук (05.22.20 – експлуатація та ремонт засобів транспорту), професор, завідувач кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України. Зауважень немає.

Офіційний опонент – Губський Андрій Миколайович, кандидат технічних наук (05.13.07 – Автоматизація процесів керування), доцент кафедри інформатики та програмної інженерії факультету інформатики і

обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Міністерства освіти і науки України.

Зауваження:

1. У першому розділі огляд практичних проектів, що використовують Go як платформу для інших мов (Grumpy, Oden, Borgo, Tengo), доцільно було б систематизувати у вигляді порівняльної таблиці за сформульованими у роботі критеріями. Це зробило б обґрунтування дослідницької ніші ще більш наочним.

2. Запропонована у другому розділі п'ятикомпонентна модель оцінювання хост-платформи має переважно якісний характер. Бажано було б запропонувати шкали або вагові коефіцієнти для окремих критеріїв, що надало б методиці прийняття рішення з підрозділу 4.12 більшої формальності та відтворюваності.

3. У розділах 2 і 4 одним із ключових критеріїв доцільності підходу виступає низька вартість реалізації нової мови, проте ця теза підтверджується переважно якісними міркуваннями. Доцільно було б навести кількісні характеристики експериментального транспілятора GoNext, зокрема обсяг власного коду та співвідношення власного і повторно використаного коду екосистеми Go, що зробило б аргумент про економіку реалізації наочнішим.

4. У підрозділі 3.13 проблему відображення позицій між вихідним кодом GoNext і згенерованим Go-кодом лише окреслено як напрям майбутнього розвитку. Бажано було б навести хоча б ескіз можливого рішення на основі механізму source maps, оскільки якість діагностики помилок безпосередньо впливає на практичну придатність транспілятора.

5. У тексті роботи зустрічаються незначні стилістичні неточності.

Офіційний опонент – Євланов Максим Вікторович, доктор технічних наук (05.13.06 – Інформаційні технології), професор, професор кафедри інформаційних управляючих систем факультету комп'ютерних наук Харківського національного університету радіоелектроніки Міністерства освіти і науки України.

Зауваження:

1. У підрозділі 2.2 виокремлено класи мов, для яких Go є доцільною ціллю транспіляції, та зазначено, що мови з радикально динамічною семантикою є менш придатними. Разом із тим цю межу окреслено стисло; доцільно було б докладніше обґрунтувати, які саме властивості Go її зумовлюють.

2. У третьому розділі описано реалізацію мовних розширень діалекту GoNext і схеми їх трансформації у Go-код. Разом із тим доцільно було б навести відомості про методику тестування транспілятора, зокрема про спосіб перевірки коректності роботи реалізованих мовних розширень, що посилює довіру до надійності експериментальної реалізації.

3. У підрозділі 3.11 взаємодію реалізованих мовних розширень проаналізовано лише на рівні окремих попарних комбінацій. Разом із тим доцільно було б навести комплексний приклад реалістичної прикладної програми, у якій усі реалізовані розширення використовуються спільно, що додатково підтвердило б практичну цілісність діалекту GoNext.

4. Запропоновану в підрозділі 4.12 покрокову методику прийняття рішення про вибір Go як цільової платформи описано достатньо повно, однак її проілюстровано фактично ретроспективно, на досвіді самого GoNext. Доцільно було б продемонструвати застосування методики до незалежного прикладу, наприклад, до вибору платформи для декларативної мови опису користувацьких інтерфейсів, подібної до SwiftUI у Swift або XAML у .NET.

Рецензент – Антоненко Світлана Валентинівна, кандидатка технічних наук (05.13.06 – автоматизовані системи управління та прогресивні інформаційні технології), доцентка, доцентка кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України.

Зауваження:

1. Автор детально аналізує обмеження класичного компактного представлення варіантів із тегом через потенційні конфлікти зі збирачем сміття та специфіку роботи з узагальненнями в Go. Водночас залишається теоретично й практично дискусійним питання: яка саме з двох інших запропонованих схем (структурна чи через `interface{}`) була врешті обрана за основу в остаточній реалізації експериментального діалекту GoNext? Які її реальні накладні витрати за пам'яттю та швидкодією у порівнянні з традиційними для Go константами (`iota`) при інтенсивних обчислювальних навантаженнях? Було б доцільно навести якісь конкретні дані порівняльного аналізу підходів.

2. У підрозділі 3.10 зазначено, що через неможливість контекстного виведення типу результату для узагальнених функцій вищого порядку короткий синтаксис анонімних функцій було обмежено лише непараметризованими контекстами. Чи розглядалися автором гнучкіші проміжні варіанти вирішення цієї проблеми (наприклад, через часткове явне вказування типів чи введення додаткових синтаксичних маркерів для складних випадків), чи повна заборона в параметризованих контекстах є єдиним прагматичним інженерним компромісом для збереження простоти транспілятора?

3. Універсальний синтаксис виклику функцій (UFCS) суттєво підвищує ергономіку коду, але створює серйозні ризики виникнення неоднозначностей. Було б доцільно показати, які конкретно детерміновані правила пріоритетності та розв'язання конфліктів імен було закладено в архітектуру транспілятора GoNext, якщо для одного й того самого типу приймача підходять декілька

різних функцій розширення, які одночасно імпортовані користувачем із різних зовнішніх пакетів.

4. Для коректної обробки іменованих аргументів та параметрів за замовчуванням транспілятор вимушений виконувати частковий семантичний аналіз та завантажувати сигнатури функцій із зовнішніх імпортованих Go-пакетів. Чи не призводить цей додатковий етап аналізу зв'язків до суттєвого сповільнення загального часу трансляції, і чи не нівелює це одну з найголовніших інженерних переваг платформи Go – традиційно високу швидкість компіляції та швидкий зворотний зв'язок?

5. У дисертації обґрунтовано, що запропонований підхід дозволяє виконувати еволюційне розширення мови Go без втрати сумісності з наявним кодом. Які саме з реалізованих у GoNext розширень (наприклад, повнофункціональні enum-типи чи іменовані аргументи) мають, на думку здобувача, найбільший потенціал і технічну готовність для включення в майбутні офіційні версії мови Go?

Рецензент – Божуха Лілія Миколаївна, кандидатка фізико-математичних наук (01.01.01 – математичний аналіз), доцентка, доцентка кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара Міністерства освіти і науки України.

Зауваження:

1. Синтаксис нових мовних конструкцій GoNext і схеми їх трансформації у розділах 2 і 3 описано переважно на рівні ілюстративних прикладів коду та словесних алгоритмів; формальної граматики розширень у роботі не наведено. Бажано було б доповнити цей опис формальним поданням — граматикою нових конструкцій, наприклад у розширеній нотації Бекуса–Наура, та правилами переписування хоча б для окремих трансформацій.

2. У підрозділі 3.5 серед можливостей конструкції match заявлено перевірку вичерпності на етапі компіляції за відсутності гілки default, проте семантичні критерії вичерпності явно не сформульовано. Зокрема, залишається незрозумілим, яка часова складність виконання match-конструкцій та як слід трактувати гілки з додатковими умовами if, які покривають варіант лише частково, та гілки, що об'єднують кілька варіантів. Варто було б навести відповідні правила явно.

3. Порівняння Go з альтернативними платформами у підрозділі 4.6 виконано на якісному рівні. На мій погляд, аргументацію посилив би наскрізний приклад: реалізація однієї й тієї самої мовної конструкції, наприклад enum-типів, поверх Go та поверх JVM або LLVM із зіставленням обсягу необхідної інфраструктури.

4. У підрозділі 4.9.1 залежність запропонованого підходу від еволюції самої мови Go лише констатовано із зауваженням, що ризик пом'якшується

строгою політикою сумісності Go. Проте для довгострокової життєздатності діалекту не менш важливим є інший аспект цієї залежності: у роботі не розглянуто стратегію супроводу транспілятора при виході нових версій хост-мови, зокрема порядок дій у разі, коли нова конструкція офіційної версії мови Go виявиться синтаксично несумісною з уже реалізованим розширенням GoNext. Бажано було б доповнити практичні рекомендації розділу 4 відповідними положеннями щодо версіонування діалекту.

5. У роботі запропоновано рішення репрезентації enum-типів на користь безпечних для збирача сміття Go структур (зокрема, з `interface{}` або структурним представленням з полями для всіх варіантів), що змусило автора відмовитися від найбільш ефективних з погляду просторової складності структур даних (зокрема, `compact tagged unions`). Вибір такого підходу спричиняє додаткові накладні витрати під час виконання програми: структурне представлення вимагає місця в пам'яті для полів усіх варіантів, навіть якщо в момент часу використовується лише один із них, а представлення через `interface{}` — непрямої адресації та можливих додаткових виділень пам'яті.

6. Задля реалізації специфічного синтаксису автором замість побудови повноцінного незалежного проміжного представлення (IR) обрано концепцію прямої трансформації синтаксичних дерев (AST), що спростило реалізацію, проте призвело до таких проблем, як ускладнення збереження точних координат токенів (`token.Pos`), розбухання AST через вираження високорівневих конструкцій (`enum`, `match`) значною кількістю базових вузлів Go та необхідність поєднувати семантичний аналіз із багатопрохідною перебудовою дерева.

Результати відкритого голосування:

«За» — 5 членів ради,

«Проти» — немає.

На підставі результатів відкритого голосування разова спеціалізована вчена рада присуджує **ФОРКЕРТУ Павлу** ступінь доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення.

Відеозапис трансляції захисту дисертації додається.

Голова разової спеціалізованої
вченої ради



(підпис)

Олег БАЙБУЗ