



канд. фіз.-мат. наук, доц. Волошко В.Л. (01.02.04 – механіка деформівного твердого тіла), доцент кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара;

д-р фіз.-мат. наук, проф. Білозьоров В.Є. (01.05.04 – системний аналіз і теорія оптимальних рішень), професор кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

д-р техн. наук, проф. Книш Л.І. (05.14.06 – технічна теплофізика та промислова теплоенергетика), професорка кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук, доц. Золотько К.Є. (05.14.04 – промислова теплоенергетика), доцент кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Зайцев В.Г. (01.05.02 – математичне моделювання та обчислювальні методи), доцент кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук Дзюба П. А. (01.02.04 – механіка деформівного твердого тіла), доцент кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Хижа О.Л. (01.01.01 – математичний аналіз), доцент кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук, доц. Мацуга О.М. (05.13.06 – автоматизовані системи управління та прогресивні інформаційні технології), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук Козакова Н. Л. (01.02.04 – механіка деформівного твердого тіла), доцентка кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара;

канд.фіз.-мат. наук Михальчук Г.Й. (01.02.04 – механіка деформівного твердого тіла), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Тонкошкур І.С. (01.02.05 – механіка рідини, газу та плазми), доцент кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук Степанова Н.І. (01.02.04 – механіка деформівного твердого тіла), доцентка кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Сафронова І.А. (01.02.04 – механіка деформівного твердого тіла), доцентка кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук, доц. Іванченко М.Г. (05.13.06 – інформаційні технології), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук, доц. Антоненко С.В. (05.13.06 – інформаційні технології), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Божуха Л.М. (01.01.01 – математичний аналіз), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук Білобородько О.І. (05.13.06 – автоматизовані системи управління та прогресивні інформаційні технології), доцентка кафедри інженерії програмного

забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. техн. наук, доц. Ємел'яненко Т.Г. (05.13.06 – автоматизовані системи управління та прогресивні інформаційні технології), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Наконечна Т.В. (01.01.01 – математичний аналіз), доцентка кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара;

канд. фіз.-мат. наук, доц. Трофімов О.В. (01.02.04 – механіка деформівного твердого тіла), доцент кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара;

доктор філософії (PhD) Антонюк В.А. (121 – інженерія програмного забезпечення), доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

доктор філософії (PhD) Єгошкін Д.І., доцент кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

Полонська А.Є., асистентка кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара;

Лисиця Н.М., асистентка кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

Сірик С.Ф., асистентка кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

Красношарпа Д.В., старший викладач кафедри комп'ютерних технологій Дніпровського національного університету імені Олеся Гончара;

Лапець О.В., асистент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

Мащенко Л.В., старший викладач кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара;

Соломатін В.А., асистент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара.

Стружко В.Р., асистент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара.

#### **ЗАПРОШЕННІ ФАХІВЦІ (4 особи, з правом голосу):**

д-р техн. наук, проф. Корчинський В.М. (05.01.01 – прикладна геометрія, інженерна графіка), професор кафедри телекомунікаційних систем та мереж факультету фізики, електроніки та комп'ютерних систем, Дніпровський національний університет імені Олеся Гончара;

д-р техн. наук, проф. Голуб С.В. (05.13.06 – інформаційні технології), завідувач кафедри програмного забезпечення автоматизованих систем факультету інформаційних технологій і систем, Черкаський державний технологічний університет;

канд. техн. наук, доц. Клименко С.В. (05.13.06 – інформаційні технології), завідувачка кафедри кібербезпеки та комп'ютерно-інтегрованих технологій, Дніпровський національний університет імені Олеся Гончара;

канд. техн. наук Губський А.М. (05.13.07 – автоматизація процесів керування), доцент кафедри інформатики та програмної інженерії факультету інформатики і обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

На засіданні присутні аспіранти: Земляний О.Д., Божуха Д.І.

## **Аспіранти участі в голосуванні не брали.**

**Порядок денний:** розгляд та обговорення дисертаційної роботи Форкерта Павла Павловича на тему «Дослідження використання рантайму Go як платформи для побудови нових мов програмування», поданої на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення.

Тема дисертації затверджена на засіданні вченої ради Дніпровського національного університету імені Олеся Гончара (протокол № 4 від 01 грудня 2022 року) та уточнена на засіданні вченої ради факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара (протокол № 6 від 23 лютого 2026 року) у формулюванні «Дослідження використання рантайму Go як платформи для побудови нових мов програмування».

Науковий керівник: кандидат технічних наук, доцент Іванченко Марина Геннадіївна.

Підготовка здобувача третього рівня вищої освіти здійснюється за акредитованою освітньо-науковою програмою «Інженерія програмного забезпечення» зі спеціальності 121 Інженерія програмного забезпечення (Сертифікат про акредитацію освітньої програми 13179, дійсний до 01.07.2030 р.).

### **СЛУХАЛИ:**

Обговорення дисертації аспіранта 4 року навчання Форкерта Павла Павловича на тему «Дослідження використання рантайму Go як платформи для побудови нових мов програмування» на здобуття ступеня доктора філософії зі спеціальності 121 Інженерія програмного забезпечення.

Перевірку на плагіат здійснювала комісія у складі: кандидат фіз.-мат. наук, доцент кафедри обчислювальної математики та математичної кібернетики Кузенков О.О., кандидат фізико-математичних наук, доцент кафедри обчислювальної математики та математичної кібернетики Козакова Н.Л., провідний інженер НДЛ ОСС Яцечко Н.Є. За результатами перевірки дисертаційної роботи на плагіат програмою «Strikeplagiarism» зроблено висновок: дисертаційна робота **Форкерта П.П.** має високий рівень унікальності (**98,33%**) і може бути допущена до захисту.

Робота виконана на **204 сторінках** і містить такі складові частини: анотація, зміст, вступ, основна частина (4 розділи), висновки, список використаних джерел, додатки.

Слово надається аспіранту Форкерту П.П. Регламент виступу – 15 хвилин.

### **Аспірант Форкерт П.П.:**

Добрий день, шановні учасники семінару. Тема моєї дисертаційної роботи: «Дослідження використання рантайму Go як платформи для побудови нових мов програмування». Науковий керівник: кандидат технічних наук, доцент Іванченко Марина Геннадіївна.

**Метою дослідження** є дослідження зручності та доцільності використання рантайму Go як платформи для побудови нових мов програмування, а також експериментальна перевірка цього підходу.

**Об'єктом дослідження** є процеси проектування та реалізації нових мов програмування і мовних розширень на базі наявних програмних платформ.

**Предметом дослідження** є методи, моделі та програмні засоби використання рантайму Go як платформи для побудови нових мов програмування або діалектів існуючих мов.

### **Основні завдання дослідження:**

1. Проаналізувати сучасні підходи до реалізації мов програмування на базі наявних платформ.
2. Дослідити особливості, специфічні для Go: його компілятор, рантайм, систему збірки та наявні інструменти аналізу й генерації коду.

3. Визначити критерії оцінювання придатності Go як хост-платформи для побудови нових мов програмування.
4. Експериментально перевірити можливість та реалістичність підходу шляхом розроблення архітектури транспілятора та діалекту GoNext.
5. Оцінити доцільність підходу загалом та окреслити межі його застосовності.

Робота містить вступ, 4 розділи, висновки, список використаних джерел, що містить 92 найменування та додатки.

Сучасна практика створення нових мов програмування дедалі рідше спирається на повну побудову компілятора, рантайму та інструментальної екосистеми з нуля. Натомість домінує підхід повторного використання вже наявних платформ, який дає змогу зменшити вартість реалізації мови, скоротити час експериментування з новими мовними конструкціями та швидше довести нову мову або діалект до практичного застосування. Найчастіше в такому контексті досліджуються JVM, LLVM, GraalVM та інші спеціалізовані платформи. Водночас придатність Go як основи для побудови нових мов програмування вивчена значно менше, хоча вона має важливі інженерні переваги: швидкий компілятор, зрілий рантайм, сучасний збирач сміття, підтримку кроскомпіляції, велику стандартну бібліотеку та розвинені засоби аналізу і генерації коду. Окремо слід зазначити, що Go не має стабільного проміжного представлення або байткоду, тому природним способом роботи з Go як платформою є саме транспіляція.

У дисертаційній роботі досліджено зручність і доцільність використання рантайму Go як платформи для побудови нових мов програмування, діалектів та предметно-орієнтованих мов через транспіляцію у звичайний Go-код. Основну увагу зосереджено на тому, які класи мовних розширень можуть бути легко і ефективно реалізовані поверх Go без модифікації офіційного компілятора, які архітектурні принципи повинен мати відповідний транспілятор, які обмеження накладає сама хост-платформа та за яких умов такий підхід є практично виправданим. Як експериментальне підтвердження запропонованого підходу розроблено діалект GoNext, який використано як демонстраційний майданчик для реалізації низки мовних розширень.

У роботі показано, що Go доцільно розглядати не як класичну багатомовну віртуальну машину і не як низькорівневий компіляторний фреймворк, а як хост-платформу особливого типу, придатну насамперед для транспільованих мов, що прагнуть зберегти сумісність із Go-екосистемою. Запропоновано систему критеріїв оцінювання такої придатності, яка охоплює виконувальні властивості, виразні можливості хост-мови, інтегрованість, інструментальну підтримку транспіляції та вартість реалізації мовних конструкцій. Також розроблено узагальнену архітектуру транспілятора до Go та класифікацію перетворень мовних розширень на синтаксичні, структурні й репрезентаційні.

У першому розділі виконано аналіз сучасних підходів до реалізації мов програмування на базі наявних платформ, зокрема JVM, LLVM та Truffle/GraalVM, і показано, що повторне використання рантаймів, компіляторних бекендів і віртуальних машин є нині домінантною тенденцією при створенні нових мов. Розглянуто роль транспіляції як самостійного механізму побудови мов, особливо доцільного тоді, коли хост-платформа не надає стабільного низькорівневого API чи байткоду, але має стабільну мову, компілятор і потужні засоби аналізу та генерації коду. Проаналізовано інженерні властивості Go, релевантні для її використання як хост-платформи, а також її інструментальну екосистему (`go/parser`, `go/ast`, `go/types`, `go/printer`, `golang.org/x/tools` тощо). Окремо розглянуто наукові й практичні роботи, дотичні до використання Go як платформи для інших мов, і встановлено, що така практика вже існує, але майже не супроводжується системним науковим аналізом.

У другому розділі сформульовано теоретичні засади використання Go як цілі транспіляції: розроблено п'ятикомпонентну систему критеріїв оцінювання, виокремлено класи мов, для яких Go є доцільною ціллю, і навпаки, для яких вона малоприматна, а також запропоновано загальну архітектуру транспілятора. Описано підходи до часткової типізації, нормалізації високорівневих конструкцій до обмеженого набору внутрішніх форм, трансформації AST у вузли go/ast та забезпечення інтероперабельності. Сформульовано класифікацію трансформацій мовних розширень і набір принципів проектування нових мов на базі Go, серед яких — мінімальна семантична відстань до Go, перевага локальних трансформацій над глобальними, максимальне використання офіційних інструментів Go та пріоритет інтероперабельності над теоретично «ідеальною» реалізацією окремої мовної конструкції.

У третьому розділі описано експериментальну реалізацію діалекту GoNext і продемонстровано практичну реалізацію повнофункціональних enum-типів, зіставлення із взірцем, значень параметрів за замовчуванням, іменованих аргументів, методів розширення, узагальнених методів, коротких анонімних функцій і універсального синтаксису виклику функцій. Кожне розширення співвіднесено із запропонованою класифікацією трансформацій: enum-типи проілюстровано як приклад репрезентаційної трансформації, match — як структурної, а ергономічні розширення — переважно як синтаксичні або структурні. Показано, що частина конструкцій потребує не лише локальних синтаксичних переписувань, а й часткового семантичного аналізу — знання сигнатур функцій, областей видимості та типів оточуючого контексту, — а також що окремі розширення утворюють змістовні комбінації, при цьому зберігаючи високу інтероперабельність зі звичайним Go, оскільки після трансформації здебільшого зникають, залишаючи звичайні Go-функції, типи та виклики у результуючому коді.

У четвертому розділі виконано комплексне оцінювання запропонованого підходу за всіма групами критеріїв — виконувальними властивостями, виразністю, інтероперабельністю, доступністю інструментів та вартістю реалізації, — визначено його межі застосовності та проведено порівняння Go з альтернативними платформами. Встановлено, що найбільший вигаш цей підхід дає для діалектів і надмножин Go, предметно-орієнтованих мов серверно-інфраструктурного профілю та нових мов, свідомо спроектованих з орієнтацією на трансформацію у Go, тоді як для мов із радикально іншою моделлю виконання, ручним керуванням пам'яттю або потребою в окремому байткодi він є малодоцільним. Сформульовано практичні рекомендації щодо проектування нових мов і транспіляторів, наведено методику прийняття рішення про вибір Go як цільової платформи та узагальнено результати на ширший клас задач, зокрема транспіляцію інших мов у Go.

У висновках представлено результати дослідження. Зазначається, що рантайм Go не є універсальною, але доцільною та прагматично виправданою платформою для побудови нових мов програмування у випадках, коли нова мова не потребує повністю автономної моделі виконання, критичною є інтеграція з наявними Go-бібліотеками, а мовні новації стосуються насамперед виразності фронтенду.

**Наукова новизна одержаних результатів полягає у наступному:**

1. **Вперше** системно обґрунтовано доцільність розгляду рантайму Go як хост-платформи особливого типу для побудови нових мов програмування не як багатомовної віртуальної машини і не як низькорівневого компіляторного фреймворку, а як цілі транспіляції для мов із високою інтероперабельністю з екосистемою Go.
2. **Удосконалено** підхід до оцінювання придатності Go як цілі транспіляції за рахунок комплексного врахування виконувальних властивостей, виразних можливостей хост-мови, інтероперабельності, інструментальної підтримки транспіляції та вартості реалізації мовних конструкцій.

3. **Удосконалено** архітектурний підхід до побудови мовних розширень у формі надмножини Go з поетапною трансформацією у звичайний Go-код, який поєднує нормалізацію конструкцій, частковий семантичний аналіз і перетворення у вузли go/ast, що дає змогу коректно реалізовувати конструкції, залежні від сигнатур функцій, областей видимості, контекстного відновлення типів і правил уніфікації типів Go, із збереженням двосторонньої інтероперабельності з екосистемою Go.
4. **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, параметрів за замовчуванням, методів розширення, узагальнених методів, коротких анонімних функцій та універсального синтаксису виклику функцій.

**Практичне значення роботи полягає у наступному:**

1. 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.
2. Розроблено експериментальний діалект GoNext та відповідний транспілятор, які підтверджують практичну здійсненність запропонованих рішень і можуть слугувати основою для подальших досліджень та прикладних розробок.
3. 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.
4. Результати дисертації впроваджено в освітній процес кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара під час викладання освітньої компоненти «Мовні технології» для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Інженерія програмного забезпечення» спеціальності 121 Інженерія програмного забезпечення.

Після закінчення доповіді до Форкерт П.П. присутніми були поставлені наступні запитання.

**ЗАПИТАННЯ ТА ВІДПОВІДІ**

**Канд. техн. наук, доцент Антоненко Світлана Валентинівна (05.13.06 – автоматизовані системи управління та прогресивні інформаційні технології), доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара**

1. Чи вірно я зрозуміла, що це підхід, коли розроблена якась нова мова програмування, Ви написали на ній код, і він проміжним парсером перебудовується у Go та відповідно компілюється стандартним компілятором Go? І для цього потрібно, щоб мовні конструкції перетворювалися на відповідні фрагменти Go-коду?
2. Чи існують метрики, за якими оцінюється повнота такої мови?
3. Чим можна проілюструвати доцільність такої роботи?

**Форкерт П.П.**

1. Так, це стандартний процес для будь-якого компілятора або інтерпретатора: виконується парсинг мови, трансформація у абстрактне синтаксичне дерево, потім його нормалізація та трансформація у синтаксичне дерево Go, з якого генерується Go-код, що потім компілюється стандартним компілятором Go.

2. Зазвичай такі метрики не збираються: процес є тюрінг-повним, тобто реалізувати можна, в принципі, будь-що. У моєму випадку я обрав набір функціональностей, яких бракує в Go і які присутні у багатьох популярних сучасних мовах програмування, і додав їх до Go, не змінюючи сам компілятор Go, у результаті чого отримав GoNext.
3. Мені хотілося отримати виразну мову на кшталт Rust, але зі збирачем сміття і не таку складну, як Rust. Rust складний саме через те, що компілятор повинен контролювати всі покажчики; якщо додати до такої мови збирач сміття, її можна суттєво спростити, і вона буде зручнішою за мову Go, яка сама по собі не є дуже виразною. Збирач сміття при цьому є частиною рантайму Go, на який я не впливаю, бо використовую його вже після того, як згенерував Go-код з іншої мови.

**Канд. фіз.-мат. наук, доцент Божуха Лілія Миколаївна (01.01.01 – математичний аналіз), доцентка кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара**

1. Перше питання стосується причин створення нових мов програмування. Чому вони виникають? З огляду на поширеність такої мови, як Python, чи потрібно створювати щось нове, чи цього вимагають певні прикладні задачі?
2. Ви говорили про синтаксичні дерева. Для побудови першого дерева та перетворення його в інше синтаксичне дерево Ви використовували структури даних, пов'язані з бібліотеками, чи розробляли власні?

**Форкерт П.П.**

1. Щодо Python: Крім Латнер, який свого часу створив LLVM, зараз працює саме над статично компільованим діалектом Python, що спеціалізується на компіляції не тільки для процесорів, а й для GPU та інших гетерогенних платформ, специфічно для задач AI-інференсу. Загалом нові мови програмування або додають функціональність, якої не було раніше, або роблять наявну функціональність зручнішою. Наприклад, з появою мови Rust керувати пам'яттю стало набагато простіше, ніж у C++, при тому, що в Rust немає збирача сміття. Тож мови програмування зазвичай створюються для того, щоб зробити розробку зручнішою і підвищити продуктивність розробників.
2. За основу взято Go AST: я частково скопіював його і додав додаткові вузли синтаксичного дерева GoNext, зокрема для enum-типів та конструкції match. Це дерево потім трансформується у стандартний Go AST, з якого Go-код генерується стандартними засобами бібліотек Go.

**Канд. техн. наук, доцент Клименко Світлана Володимирівна (05.13.06 – інформаційні технології), завідувачка кафедри кібербезпеки та комп'ютерно-інтегрованих технологій Дніпровського національного університету імені Олеся Гончара**

1. У висновках ми звикли бачити результати ґрунтового аналізу за мовами, проте аналіз стосується переважно мови Go. Чому про інші мови нічого не сказано?
2. Зазначено, що експериментально підтверджено 8 мовних розширень. Чому недостатньо було одного, і які саме функції вони охоплюють для вирішення задач такого типу?
3. Світ дуже швидко рухається, і ми очікуємо збільшення швидкості процесів обробки. Чи ставилася така задача у дисертації, наприклад, для використання на серверних пристроях?

**Форкерт П.П.**

1. Щодо транспіляції саме у мову Go наукових публікацій практично немає. Існують три практичні проекти, які на певному етапі були активними: Grumpy – транспілятор з Python у Go, Borgo – нова мова з Rust-подібним синтаксисом, що

компілюється в Go, та Oden – повністю нова функціональна мова на базі Go, проект якої був закритий кілька років тому. Тому оцінювати щось з точки зору публікацій тут практично неможливо, і огляд літератури значно більшою мірою стосувався віртуальних машин, LLVM та транспіляції у TypeScript – найпопулярнішого транспілятора у світі.

2. У дисертації трансформації мовних розширень поділено на три види: синтаксичні – локальні переписування у синтаксичному дереві; структурні – зміни, що зачіпають кілька місць дерева; та репрезентаційні – ті, що змінюють саме представлення даних у згенерованому Go-кодi (як enum-типи). Частина конструкцій до того ж потребує не лише змін у дереві, а й часткового семантичного аналізу — інформації про сигнатури функцій, типи та області видимості. Вісім реалізованих розширень охоплюють усі три види трансформацій, а окремі розширення логічно потягнули за собою інші – наприклад, enum-типи потягнули за собою match.
3. Я мав на увазі той факт, що GoNext можна використовувати замість Go, бо він повністю сумісний з Go і просто додає додаткову функціональність та зручність, наприклад, enum-типи та match. І його, як і сам Go, звичайно можна використовувати на серверних пристроях.

**Доктор філософії (PhD) Антонюк Владислав Андрійович (121 – інженерія програмного забезпечення), доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара**

1. Чи можна замість Go конвертувати Вашу розробку в іншу мову програмування, наприклад .NET, чи вона конвертує тільки у Go?
2. Зазвичай мова створюється під конкретні потреби. GoNext буде фактично використовуватись там само, де використовується Go. Чи має сенс створювати такі надбудови, чи доцільніше було б контриб'юити безпосередньо у саму мову Go і додати туди ті ж самі enum та інше?
3. Чи буде GoNext заточений під конкретну версію Go?
4. Що мені потрібно, щоб почати працювати саме з GoNext?

**Форкерт П.П.**

1. Ні, рішення специфічне для Go: наприклад, представлення enum-типів обиралося саме з урахуванням особливостей Go, і вся робота націлена на оцінку Go як платформи для інших мов програмування. JVM зазвичай використовує набагато більше пам'яті, ніж типові програми на Go, а Go дуже популярний у серверній розробці, що видно з екосистеми Docker і Kubernetes, тому доцільно зробити більш комфортну мову саме для цієї екосистеми. Крім того, віртуальні машини потребують встановлення додаткового програмного забезпечення для запуску програм та мають проблему повільного старту через JIT-компіляцію, тоді як Go компілюється у бінарний файл з машинним кодом, який можна просто запустити, що суттєво спрощує розгортання програм.
2. Якщо є можливість, треба контриб'юити в існуючі проекти, але вірогідність того, що подібні зміни приймуть у сам Go, надзвичайно мала: Go має дуже жорсткі вимоги сумісності з попереднім кодом, і нову функціональність розробники Go додають дуже неохоче – на додавання дженеріків пішло близько 10 років, а узагальнені методи з додатковими типовими параметрами у стандартному Go неможливо реалізувати досі. Завдяки тому, як у GoNext реалізовано методи розширення, такі узагальнені методи стали можливими. Тобто отримати подібну функціональність, намагаючись контриб'юити у стандартний репозиторій Go, практично неможливо або на це підуть роки. Транспіляція ж дозволяє рухатися

швидше, перевіряти нові ідеї та розуміти, у який Go-код вони трансформуються, що корисно і для потенційних пропозицій у сам Go.

3. Зараз GoNext використовує той компілятор Go, який встановлено на машині. Одним з ризиків підходу є те, що зміни у Go можуть вплинути на згенерований код, але цей ризик суттєво пом'якшується політикою сумісності Go: за понад 10 років існуючі програми на Go практично ніколи не ламалися, у тому числі великі.
4. Потрібен лише встановлений компілятор Go.

**Д-р техн. наук, професор Байбуз Олег Григорович (05.22.20 – експлуатація та ремонт засобів транспорту), завідувач кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара**

Усе ж таки до чого тут наука? Що саме тут наукового, докторського, філософського? За якими критеріями можна зрозуміти, що це наукова робота, а не суто інженерна чи технологічна?

**Форкерт П.П.**

Науковим результатом роботи є не сама програмна реалізація, а обґрунтування підходу: системне дослідження придатності Go-рантайму як платформи для нових мов, запропоновані критерії оцінювання такої придатності та методи трансформації мовних конструкцій.

Якщо подивитися ширше, існують прецеденти наукових робіт за подібною тематикою: дисертації та статті про JIT-компіляцію, JVM, LLVM; зокрема Кріс Латтнер свою дисертацію присвятив саме LLVM, який він і розробив в університеті. Тож подібні роботи дозволяють формалізувати та сфокусувати майбутні напрями досліджень і практичних розробок у певному руслі, і без подібних дисертацій та статей частина аплікацій, мов програмування чи інфраструктурних систем могла б взагалі не з'явитися.

**Канд. техн. наук, доцент Антоненко Світлана Валентинівна, доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара**

А коли йдеться про економію коштів на розробці мов програмування, чи можна це кількісно показати: з чого складається ціна і де саме можна зекономити?

**Форкерт П.П.**

Реалізувати окремі компоненти можливо, але дуже складно оцінити коректність та надійність, наприклад, конкурентного рантайму, який у Go перевірений роками на великих проєктах та великій кількості серверів. Саме це одна з причин, чому обрано Go: він має зрілий рантайм, який не потребує віртуальної машини, та дуже якісний збирач сміття, перевірений часом. Як коректно квантифікувати це у коштах, сказати складно.

**Голова семінару, член-кореспондент НАН України, доктор фізико-математичних наук, проф. Кісельова О.М., декан факультету прикладної математики та інформаційних технологій, професор кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара:**

Запитань більше немає. Переходимо до обговорення дисертаційної роботи. Слово має науковий керівник.

**ВИСТУП НАУКОВОГО КЕРІВНИКА:**

Дисертаційна робота Форкерта Павла Павловича присвячена дослідженню використання рантайму Go як платформи для побудови нових мов програмування. Робота спрямована на зменшення вартості та складності створення нових мов і мовних розширень за рахунок транспіляції у Go, що дозволяє використовувати зрілий компілятор, рантайм та

екосистему цієї мови без розроблення власного бекенду. У роботі запропоновано систему критеріїв оцінювання придатності Go як цільової платформи, розроблено архітектуру транспілятора та створено експериментальний діалект GoNext, який підтверджує можливість реалізації сучасних мовних конструкцій поверх Go без модифікації її офіційного компілятора.

Мені пощастило бути науковим керівником Павла Павловича. Окрім того, що він є висококваліфікованим фахівцем зі значним практичним досвідом роботи в індустрії, він є людиною, яка, якщо за щось береться, робить це дуже якісно, занурюючись у деталі, вивчаючи суміжні питання та глибоко опрацьовуючи підґрунтя тих питань, за які береться. Питаннями специфіки та порівняння існуючих мов програмування Павло цікавиться досить давно, тож це близька для нього тематика: йому було цікаво, наскільки ефективними є існуючі рішення, що цікавого є у тій чи іншій мові, що можна покращити чи створити нового.

Вибір теми дослідження та вся пророблена робота були здійснені Павлом самостійно, отримані результати є новими і були представлені науковій спільноті на конференціях та у фахових наукових журналах. Робота велася систематично: щодо формальних вимог виконання навчального плану аспірантів питань не було. Павло працював над роботою сумлінно і зацікавлено, вивчаючи наявні дисертаційні роботи та публікації не формально, а з дійсним інтересом до того, що існує зараз і як це можна покращити.

Вважаю, що дисертаційна робота Павла «Дослідження використання рантайму Go як платформи для побудови нових мов програмування» є завершеним науковим дослідженням, яке містить нові теоретично обґрунтовані та практично значущі результати. Робота виконана на високому науковому рівні, має актуальність, новизну та практичну цінність. Застосовані методи дослідження відповідають сучасним вимогам. Як науковий керівник рекомендую роботу до захисту на здобуття ступеня доктора філософії зі спеціальності 121 Інженерія програмного забезпечення.

Дякую за увагу.

### **В ОБГОВОРЕННІ ДИСЕРТАЦІЇ ФОРКЕРТА П.П. ВЗЯЛИ УЧАСТЬ:**

**Канд. техн. наук, доцент Антоненко Світлана Валентинівна – доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара:**

Робота дуже цікава і відповідає нашій спеціальності; мені здається, що саме таких робіт нам бажано мати більше. Я підтримую цю роботу, буду уважно вивчати її текст та підтверджую готовність бути рецензентом.

**Канд. фіз.-мат. наук, доцент Божуха Лілія Миколаївна – доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара:**

Я підтримую цю роботу. Представлений на семінарі матеріал переконливо свідчить, що дослідження повністю відповідає спеціальності 121 Інженерія програмного забезпечення. Відповіді Павла на запитання, зокрема щодо причин створення нових мов програмування та структур даних, використаних для роботи з синтаксичними деревами, були ґрунтовними і продемонстрували глибоке володіння матеріалом. Зі свого боку підтверджую можливість бути рецензентом цієї роботи.

**Доктор філософії (PhD) Антонюк Владислав Андрійович – доцент кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара:**

З огляду на питання, що я ставив, у мене спочатку були певні сумніви щодо доцільності створення подібної роботи, оскільки вже існує досить велика кількість мов програмування, можливо, під усі наявні технології та потреби. Але беручи до уваги відповіді та дивлячись на роботу з точки зору саме дослідження, тобто того, що вимагається від робіт на здобуття ступеня доктора філософії, дослідження проведено, робота є актуальною, мова Go є дуже популярною, тому зауважень до роботи в мене немає.

## **ВИСНОВОК**

**Актуальність теми дисертації.** У сучасній інженерії програмного забезпечення створення нових мов програмування є не лише теоретичною проблемою, а й практичною інженерною задачею. Класичний підхід до створення мови передбачає повний цикл робіт – від формалізації граматики до розроблення власного рантайму, збирача сміття та супровідних інструментів, що є надзвичайно дорогим за часом і складністю. Саме тому абсолютна більшість сучасних мов програмування створюється не «з нуля», а на базі вже наявних платформ, таких як JVM, LLVM, GraalVM тощо. Водночас мова Go, яка набула широкого поширення у хмарній, серверній та інфраструктурній розробці програмного забезпечення і має потужну екосистему інструментів для аналізу та генерації коду, як платформа для побудови нових мов програмування досліджена значно менше.

Актуальність цього дослідження визначається зростанням інтересу до нових мов програмування як засобу підвищення продуктивності розробників, потребою у дешевших методах створення нових мов, які не вимагають повної реалізації власного рантайму, поширеністю Go у серверній та інфраструктурній розробці, наявністю у Go розвинених інструментів для аналізу й генерації коду, а також потребою окреслити межі застосовності транспіляції у Go як альтернативи компіляції у байткод або машинний код. Таким чином, актуальною науково-технічною задачею є розроблення та обґрунтування методології використання рантайму Go як платформи для побудови нових мов програмування, включно з діалектами, предметно-орієнтованими мовами та транспільованими надмножинами існуючих мов.

**Затвердження теми та плану дисертації.** Тема дисертації затверджена на засіданні вченої ради Дніпровського національного університету імені Олеся Гончара (протокол № 4 від 01 грудня 2022 року) та уточнена на засіданні вченої ради факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара (протокол № 6 від 23 лютого 2026 року).

Науковим керівником дисертаційної роботи призначено кандидата технічних наук, доцента Іванченко Марину Геннадіївну.

Підготовка здобувача третього рівня вищої освіти здійснювалась за акредитованою освітньо-науковою програмою «Інженерія програмного забезпечення» зі спеціальності 121 Інженерія програмного забезпечення (сертифікат про акредитацію освітньої програми № 13179, дійсний до 01.07.2030 р.).

**Зв'язок роботи з науковими програмами, планами, темами.** Дисертаційна робота виконувалась у відповідності з індивідуальним планом підготовки аспіранта кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара та пов'язана з дослідженнями в галузі інженерії програмного забезпечення, технологій компіляції, мов програмування та програмних платформ.

Дослідження здійснювалось в рамках науково-дослідної роботи № ФПМ-2-22 «Розроблення програмного забезпечення аналізу та кластеризації часових рядів» 2022-2024

рр. номер держреєстрації 0122U001465 та № 58 – ФПМ-2-25 «Розроблення інформаційної технології обробки статистичних даних» 2025-27 рр. номер держреєстрації 0125U002280.

**Публікації та особистий внесок здобувача.** За темою дисертації опубліковано 11 робіт: 3 статті у наукових фахових виданнях України категорії Б та 8 тез доповідей у збірниках матеріалів наукових конференцій. Аналіз літературних джерел, розроблення архітектури транспілятора, реалізація експериментального діалекту GoNext та мовних розширень, проведення експериментів, обробка та аналіз отриманих результатів виконані автором самостійно. Постановка мети і завдань дослідження, аналіз і узагальнення отриманих результатів проводились спільно з науковим керівником к.т.н., доц. М. Г. Іванченко. Публікації Форкерта П.П. відповідають вимогам пп. 8, 9 «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 р. № 44 (зі змінами).

**Обґрунтованість наукових положень, висновків і рекомендацій.** Обґрунтованість наукових положень, висновків та рекомендацій дисертації забезпечується коректністю постановки задач, використанням апробованих методів теорії мов програмування, компіляторобудування та інженерії програмного забезпечення, експериментальною перевіркою запропонованих підходів на реалізації діалекту GoNext, а також узгодженістю отриманих результатів з раніше опублікованими дослідженнями інших авторів, що підтверджує їхню достовірність та несуперечливість.

**Наукова новизна одержаних результатів** полягає у наступному:

1. **Вперше** системно обґрунтовано доцільність розгляду рантайму Go як хост-платформи особливого типу для побудови нових мов програмування не як багатомовної віртуальної машини і не як низькорівневого компіляторного фреймворку, а як цілі транспіляції для мов із високою інтероперабельністю з екосистемою Go.
2. **Удосконалено** підхід до оцінювання придатності Go як цілі транспіляції за рахунок комплексного врахування виконувальних властивостей, виразних можливостей хост-мови, інтероперабельності, інструментальної підтримки транспіляції та вартості реалізації мовних конструкцій.
3. **Удосконалено** архітектурний підхід до побудови мовних розширень у формі надмножини Go з поетапною трансформацією у звичайний Go-код, який поєднує нормалізацію конструкцій, частковий семантичний аналіз і перетворення у вузли go/ast, що дає змогу коректно реалізовувати конструкції, залежні від сигнатур функцій, областей видимості, контекстного відновлення типів і правил уніфікації типів Go, із збереженням двосторонньої інтероперабельності з екосистемою Go.
4. **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, параметрів за замовчуванням, методів розширення, узагальнених методів, коротких анонімних функцій та універсального синтаксису виклику функцій.

**Практичне значення роботи** полягає у наступному:

1. 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.
2. Розроблено експериментальний діалект GoNext та відповідний транспілятор, які підтверджують практичну здійсненність запропонованих рішень і можуть слугувати основою для подальших досліджень та прикладних розробок.

3. 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.
4. Результати дисертації впроваджено в освітній процес кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара під час викладання освітньої компоненти «Мовні технології» для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Інженерія програмного забезпечення» спеціальності 121 Інженерія програмного забезпечення.

#### Список опублікованих праць за темою дисертації

##### *Статті у наукових фахових виданнях України:*

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> (особистий внесок Форкerta П.П.: провів аналіз підходів до реалізації мов програмування та можливостей використання транспіляції в Go, сформулював вимоги до повнофункціональних епит-типів у діалекті GoNext, запропонував синтаксис їх оголошення, дослідив та порівняв кілька варіантів представлення епит-типів у Go-кодi, зокрема з урахуванням узагальнень, вказівників, роботи збирача сміття та накладних витрат, розробив підхід до генерації конструкторів, Match-методів і допоміжних методів доступу, а також проаналізував сумісність запропонованого рішення з екосистемою Go; Іванченко М.Г.: постановка завдання, узагальнення отриманих результатів).
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>. (особистий внесок Форкerta П.П.: провів порівняльний аналіз підходів до реалізації іменованих аргументів у сучасних мовах програмування; обґрунтував вибір call-site-підходу, за якого іменовані аргументи не потребують змін у визначеннях функцій, запропонував синтаксис іменованих аргументів для GoNext, дослідив його сумісність із граматикою Go та розробив алгоритм транспіляції, що передбачає визначення сигнатури функції, перевірку помилкових і дубльованих імен параметрів, окремо проаналізував інтеграцію іменованих аргументів із механізмом значень параметрів за замовчуванням; Іванченко М.Г.: постановка мети дослідження, контроль та узагальнення отриманих результатів).
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>. (особистий внесок Форкerta П.П.: провів аналіз реалізацій методів розширення в сучасних мовах програмування та порівняв їх з універсальним синтаксисом виклику функцій, сформулював вимоги до реалізації методів розширення в GoNext з урахуванням мінімальних накладних витрат і повної сумісності зі стандартним Go, запропонував синтаксис оголошення методів розширення через модифікатор *extension*, механізм *import extension* для використання функцій наявних Go-бібліотек як методів розширення, а також алгоритм переписування викликів виду *value.Method(args)* у звичайні виклики функцій, дослідив застосування цього підходу для підтримки узагальнених методів, які відсутні у стандартному Go,

та проаналізував інтеперабельність запропонованого рішення з існуючим кодом на Go; Іванченко М.Г.: постановка задачі, аналіз результатів).

**Наукові праці, які засвідчують апробацію матеріалів дисертації:**

4. Forkert P. P., Sydorova M. G., Honcharova Yu. S. MODERN ARCHITECTURE OF DYNAMICALLY TYPED PROGRAMMING LANGUAGE VMS. *Тези доповідей IV Всеукраїнської науково-практичної конференції молодих науковців та здобувачів вищої освіти «Сучасні науково-технічні дослідження у контексті мовного простору»*, Дніпро, 11 травня 2023, с. 188-190. URL: <https://www.confcontact.com/2023-suchasni-ntd/3-Forkert-Sydorova-Honcharova.pdf> (особистий внесок Форкерта П.П.: провів аналіз архітектур сучасних віртуальних машин динамічно типізованих мов програмування, зокрема V8, SpiderMonkey та JavaScriptCore, узагальнив спільну для них багаторівневу схему виконання з інтерпретатором і JIT-компіляторами; Іванченко М.Г.: постановка завдання, узагальнення результатів; Гончарова Ю.С.: аналіз формулювань).
5. Forkert P. P., Sydorova M. G. ADVANTAGES OF GOLANG AS A FOUNDATION FOR NEW PROGRAMMING LANGUAGES. *Тези доповідей XXI міжнародної науково-практичної конференції «МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ (МПЗІС-2023)»*, Дніпро, 22-24 листопада 2023, с. 7-8. URL: <https://mpzis.dnu.dp.ua/wp-content/uploads/2023/11/mpzis-2023.pdf> (особистий внесок Форкерта П.П.: проаналізував переваги Go як основи для побудови нових мов програмування, зокрема продуктивність рантайму, підтримку конкурентності, збирач сміття, кроскомпіляцію та екосистему бібліотек, обґрунтував транспіляцію як основний спосіб використання Go-рантайму гостьовими мовами; Іванченко М.Г.: постановка завдання, узагальнення результатів).
6. Forkert P. P., Sydorova M. G. CHALLENGES OF USING GOLANG AS A FOUNDATION FOR NEW PROGRAMMING LANGUAGES. *Тези доповідей VI Всеукраїнської науково-практичної інтернет-конференція студентів, аспірантів та молодих вчених «Сучасні інформаційні системи та технології»*, Хмельницький, 30 листопада 2023, с. 55-56. URL: <https://kntu.net.ua/ukr/content/download/110490/623634/file/CICT2023.pdf> (особистий внесок Форкерта П.П.: систематизував обмеження використання Go як хост-платформи, зокрема відсутність стабільного API рантайму та готових інструментів транспіляції, обмеженість узагальнень, і запропонував шляхи їх подолання; Іванченко М.Г.: постановка завдання, узагальнення результатів).
7. Forkert P. P., Sydorova M. G. IMPROVING ENUMS IN GO PROGRAMMING LANGUAGE DIALECT. *Тези доповідей VI Міжнародної науково-практичної конференції молодих вчених та студентів «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ПЕРЕДОВІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ (SOFT TECH-2024)»*, Київ, 21-23 травня 2024, с. 148-150. URL: <https://drive.google.com/file/d/18bZ9QBure7U08rbqiHmxWglTrK1C9D4L/view> (особистий внесок Форкерта П.П.: запропонував для GoNext синтаксис зіставлення із взірцем match, адаптований до стилю оператора switch у Go, із захисними умовами та перевіркою вичерпності, а також схему його транспіляції у виклики згенерованих Match-методів enum-типів; Іванченко М.Г.: постановка завдання, аналіз результатів).
8. Форкерт П. П., Іванченко М. Г. УНІВЕРСАЛЬНИЙ СИНТАКСИС ВИКЛИКУ ФУНКЦІЙ В ДІАЛЕКТІ МОВИ ПРОГРАМУВАННЯ GO. *Збірник наукових праць*

- IV Міжнародної науково-практичної конференції «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ ТА НАУЦІ», Запоріжжя, 20 травня 2025, с. 561-565 (особистий внесок Форкерта П.П.: проаналізував реалізації універсального синтаксису виклику функцій у сучасних мовах програмування, запропонував варіант його інтеграції в GoNext зі збереженням зворотної сумісності з наявним Go-кодом та правила розв'язання неоднозначностей під час пошуку функцій; Іванченко М.Г.: постановка завдання, узагальнення результатів).*
9. Форкерт П. П., Іванченко М. Г. КОРОТКИЙ СИНТАКСИС ДЛЯ АНОНІМНИХ ФУНКЦІЙ В ДІАЛЕКТІ МОВИ ПРОГРАМУВАННЯ GO. *Тези доповідей VI Міжнародної науково-практичної інтернет конференції «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: МОДЕЛІ, АЛГОРИТМИ, СИСТЕМИ (ITMAS – 2025)», Миколаїв, 16-17 листопада 2025, с. 531-533. URL: <https://itconf.nuos.edu.ua/2025/publications/short-syntax-for-anonymous-functions-in-go-programming-language-dialect/> (особистий внесок Форкерта П.П.: запропонував розширення граматики GoNext короткими літералами анонімних функцій та підхід до їх транспіляції з відновленням пропущених типів шляхом уніфікації з очікуваним типом функцій; Іванченко М.Г.: постановка завдання, контроль результатів).*
10. Forkert P. P., Ivanchenko M. G. IMPLEMENTING DEFAULT PARAMETER VALUES IN GO PROGRAMMING LANGUAGE DIALECT. *Тези доповідей XXIII міжнародної науково-практичної конференції «МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ (МПЗІС-2025)», Дніпро, 19-21 листопада 2025, с. 8-9. URL: <https://mpzis.dnu.dp.ua/wp-content/uploads/2025/11/%D0%9C%D0%9F%D0%97%D0%86%D0%A1-2025.pdf> (особистий внесок Форкерта П.П.: запропонував синтаксис значень параметрів за замовчуванням для GoNext та алгоритм їх транспіляції через генерацію допоміжних функцій з опціональними параметрами, що коректно обробляє значення, залежні від області видимості означення функції; Іванченко М.Г.: постановка завдання, аналіз результатів).*
11. Forkert P. P., Ivanchenko M. G. A GENERALIZED TRANSPILER ARCHITECTURE FOR LANGUAGES TARGETING GO. *Тези доповідей XXVI всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ», Одеса, 16-17 квітня 2026, с. 143-145. URL: [https://drive.google.com/file/d/1QEcwdAJ9J4nzSCRZZ5X\\_eW5JFQE7a5vh/view](https://drive.google.com/file/d/1QEcwdAJ9J4nzSCRZZ5X_eW5JFQE7a5vh/view) (особистий внесок Форкерта П.П.: описав узагальнену багатоетапну архітектуру транспілятора для мов, що компілюються в Go, виокремив нормалізацію конструкцій як ключовий етап, що забезпечує модульність; Іванченко М.Г.: постановка завдання, узагальнення результатів).*

**На підставі заслуховування та обговорення доповіді Форкерта П.П. про основні положення дисертаційної роботи, питань та відповідей на них**

#### **УХВАЛИЛИ:**

1. Вважати, що за актуальністю, ступенем новизни, обґрунтованості, наукової та практичної цінності здобутих результатів дисертація Форкерта Павла Павловича на тему «Дослідження використання рантайму Go як платформи для побудови нових мов програмування» відповідає вимогам Порядку присудження ступеня доктора філософії та скасування рішення разової спеціалізованої вченої ради закладу вищої освіти, наукової установи про присудження ступеня доктора філософії від 12 січня 2022 р. № 44 (зі змінами).

2. Рекомендувати дисертаційну роботу Форкерт Павла Павловича на тему «Дослідження використання рантайму Go як платформи для побудови нових мов програмування» до захисту в разовій спеціалізованій вченій раді на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення.

3. Клопотати перед вченою радою університету розглянути питання про створення спеціалізованої вченої ради для проведення разового захисту дисертації на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення Форкерт Павла Павловича у такому складі:

№ з/п	Прізвище, ім'я, по батькові	Місце основної роботи, підпорядкування, посада	Науковий ступінь, шифр, назва спеціальності, за якою захищена дисертація, рік присудження	Вчене звання (за спеціальністю, кафедрою), рік присвоєння	Наукові публікації, опубліковані за останні п'ять років, за науковим напрямом, за яким підготовлено дисертацію здобувача, з яких не менше однієї публікації у виданнях, проіндексованих у базах даних Scopus та/або Web of Science Core Collection
1	2	3	4	5	6
1.	Байбуз Олег Григорович (голова)	Дніпровський національний університет імені Олеся Гончара Міністерства освіти і науки України, завідувач кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій	Доктор технічних наук, 05.22.20 – експлуатація та ремонт засобів транспорту, 2004 рік	Професор за кафедрою математичного забезпечення електронних обчислювальних машин, 2007 рік	1. Божуха Д. І., Байбуз О. Г. Дослідження моделей хмарних систем на прикладі найпростішого потоку. <i>Актуальні проблеми автоматизації та інформаційних технологій</i> . 2025, no. 29. Р. 84–91. DOI: <a href="http://dx.doi.org/10.15421/432508">http://dx.doi.org/10.15421/432508</a> URL: <a href="https://actualproblems.dp.ua/index.php/APAIT/article/view/297">https://actualproblems.dp.ua/index.php/APAIT/article/view/297</a> (фахове видання, категорія «Б» 115) 2. Вакульчик С. О., Байбуз О. Г. Аналіз та реалізація методу підтримки прийняття рішень. <i>Актуальні проблеми автоматизації та інформаційних технологій</i> . 2024, no. 28. Р. 265–273. DOI: <a href="http://dx.doi.org/10.15421/432425">http://dx.doi.org/10.15421/432425</a> URL: <a href="https://actualproblems.dp.ua/index.php/APAIT/article/view/270">https://actualproblems.dp.ua/index.php/APAIT/article/view/270</a> (фахове видання, категорія «Б» 115) 3. Божуха Д. І., Байбуз О. Г., Машенко Л. В. Про підходи дослідження системи хмарних обчислень. <i>Актуальні проблеми автоматизації та інформаційних технологій</i> . 2022, no. 26. Р. 18–30. DOI: <a href="http://dx.doi.org/10.15421/432203">http://dx.doi.org/10.15421/432203</a> URL: <a href="https://actualproblems.dp.ua/index.php/APAIT/article/view/211">https://actualproblems.dp.ua/index.php/APAIT/article/view/211</a> (фахове видання, категорія «Б» 115)
2	Губський Андрій Миколайович (опонент)	Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Міністерства освіти і науки України,	Кандидат технічних наук, 05.13.07 — Автоматизація процесів керування, 2017 рік		1. Губський А.М. Метод імпутовання пропусків у бінарних даних. <i>Актуальні проблеми автоматизації та інформаційних технологій</i> , 2025. Том 29. С. 179–187. DOI: <a href="http://dx.doi.org/10.15421/432516">http://dx.doi.org/10.15421/432516</a> URL: <a href="https://actualproblems.dp.ua/index.php/APAIT/article/view/305">https://actualproblems.dp.ua/index.php/APAIT/article/view/305</a> (фахове видання, категорія «Б» 115) 2. Марченко О. І., Корнага Я. І., Губський А. М. Динамічний механізм стійкості для масштабованих інформаційних інфраструктур. <i>Вісник Херсонського</i>

		доцент кафедри інформатики та програмної інженерії, факультету інформатики і обчислювальної техніки			<p>національного технічного університету, 2025. Том 2 № 1(92). С. 60–65. DOI: <a href="https://doi.org/10.35546/kntu2078-4481.2025.1.2.8">https://doi.org/10.35546/kntu2078-4481.2025.1.2.8</a> URL: <a href="https://journals.kntu.kherson.ua/index.php/visnyk_kntu/article/view/852">https://journals.kntu.kherson.ua/index.php/visnyk_kntu/article/view/852</a> (фахове видання, категорія «Б» 373)</p> <p>3. Дзівідзінська М., Фіногенов О., Губський А. Інтеграція систем з використанням каталогу довідників. <i>Адаптивні системи автоматичного управління</i>, 2024. Том 1 № 44. С. 62–70. DOI: <a href="https://doi.org/10.20535/1560-8956.44.2024.302418">https://doi.org/10.20535/1560-8956.44.2024.302418</a> URL: <a href="https://asac.kpi.ua/article/view/302418">https://asac.kpi.ua/article/view/302418</a> (фахове видання, категорія «Б», 732)</p>
3	Євланов Максим Вікторович (опонент)	Харківський національний університет радіоелектроніки Міністерства освіти і науки України, професор кафедри інформаційних управляючих систем факультету комп'ютерних наук	Доктор технічних наук, 05.13.06 – Інформаційні технології, 2017 рік	Професор по кафедрі інформаційних управляючих систем, 2023 рік	<p>1. Кожанов А. Є., Євланов М. В., Мороз Д. М. Вдосконалення методу вирішення задачі ранньої ідентифікації конфігураційних елементів інформаційної системи. <i>Системні технології</i>, 2026. Том 3(164). С. 59-72. DOI: <a href="https://doi.org/10.34185/1562-9945-3-164-2026-05">https://doi.org/10.34185/1562-9945-3-164-2026-05</a> URL: <a href="https://journals.nmetau.edu.ua/index.php/st/uk/article/view/2317">https://journals.nmetau.edu.ua/index.php/st/uk/article/view/2317</a> (фахове видання, категорія «Б» 1147)</p> <p>2. Круглик А. С., Левикін В. М., Євланов М. В., Мороз Б. І., Мороз Д. М. Модифікація моделі одномісного коваріантного функтора для процесу кросплатформної міграції інформаційної системи. <i>Автоматизовані системи управління та прилади автоматики</i>, 2025. Том 185. С. 5-23. DOI: <a href="https://doi.org/10.30837/0135-1710.2025.185.005">https://doi.org/10.30837/0135-1710.2025.185.005</a> URL: <a href="https://asu-pa.nure.ua/article/view/336285">https://asu-pa.nure.ua/article/view/336285</a> (фахове видання, категорія «Б» 394)</p> <p>3. Євланов М. В., Мороз Б. І., Мороз Д. М., Лучицький В. В. Інформаційна технологія виявлення термінів та артефактів проекту у вимогах до інформаційної системи. <i>Автоматизовані системи управління та прилади автоматики</i>, 2024. Том 182. С. 73-93. DOI: <a href="https://doi.org/10.30837/0135-1710.2024.182.073">https://doi.org/10.30837/0135-1710.2024.182.073</a> URL: <a href="https://asu-pa.nure.ua/article/view/317946">https://asu-pa.nure.ua/article/view/317946</a> (фахове видання, категорія «Б» 394)</p>
4	Антоненко Світлана Валентинівна (рецензент)	Дніпровський національний університет імені Олеся Гончара Міністерства освіти і науки України, доцент кафедри інженерії програмного	Кандидат технічних наук, 05.13.06 – автоматизовані системи управління та прогресивні інформаційні	Доцент за кафедрою математичного забезпечення електронних обчислювальних машин, 2004 рік	<p>1. Новіков С. О., Антоненко С. В., Ізмайлова М. К. Порівняння методів передачі даних у розподілених системах з мобільними інтерфейсами. <i>Актуальні проблеми автоматизації та інформаційних технологій</i>, 2025. Том 29. С. 331-342. DOI: <a href="http://dx.doi.org/10.15421/432530">http://dx.doi.org/10.15421/432530</a> URL: <a href="https://actualproblems.dp.ua/index.php/AP">https://actualproblems.dp.ua/index.php/AP</a></p>

		забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій	і технології, 2001 рік		<p><a href="#">АІТ/article/view/319</a> (фахове видання, категорія «Б» 115)</p> <p>2. Соколянська Т. О., Антоненко С. В. Проектування та розроблення мобільної платформи helphub для координації волонтерської діяльності. <i>Актуальні проблеми автоматизації та інформаційних технологій</i>, 2025. Том 29. С. 370-381. DOI: <a href="http://dx.doi.org/10.15421/432534">http://dx.doi.org/10.15421/432534</a> URL: <a href="https://actualproblems.dp.ua/index.php/AP">https://actualproblems.dp.ua/index.php/AP</a></p> <p><a href="#">АІТ/article/view/323</a> (фахове видання, категорія «Б» 115)</p> <p>3. Ганжа А. С., Антоненко С. В. Автоматизація розробки WIN32 десктопних додатків: практичні підходи і стратегії. <i>Системні технології</i>, 2025. Том 1(156). С. 12-22. DOI: <a href="https://doi.org/10.34185/1562-9945-1-156-2025-02">https://doi.org/10.34185/1562-9945-1-156-2025-02</a> URL: <a href="https://journals.nmetau.edu.ua/index.php/st/uk/article/view/1939">https://journals.nmetau.edu.ua/index.php/st/uk/article/view/1939</a> (фахове видання, категорія «Б» 1147)</p>
5	Божуха Лілія Миколаївна (рецензент)	Дніпровський національний університет імені Олеся Гончара Міністерства освіти і науки України, доцент кафедри інженерії програмного забезпечення та інформаційних технологій факультету прикладної математики та інформаційних технологій	Кандидат фізико-математичних наук, 01.01.01 – математичний аналіз, 2003 рік	Доцент за кафедрою програмного забезпечення та обчислювальної техніки, 2006 рік	<p>1. Логвин Д. А., Божуха Л. М. Метрики оцінювання ефективності програмного забезпечення для обробки природної мови. <i>Актуальні проблеми автоматизації та інформаційних технологій</i>. 2025, no. 29. Р. 266–276. DOI: <a href="http://dx.doi.org/10.15421/432524">http://dx.doi.org/10.15421/432524</a> URL: <a href="https://actualproblems.dp.ua/index.php/AP">https://actualproblems.dp.ua/index.php/AP</a></p> <p><a href="#">АІТ/article/view/313</a> (фахове видання, категорія «Б» 115)</p> <p>2. Сімакін С. К., Божуха Л. М. Метод інтелектуального керування часом життя кешу вебсервісів на основі алгоритмів навчання з підкріпленням. <i>Актуальні проблеми автоматизації та інформаційних технологій</i>. 2025, no. 29. Р. 363–369. DOI: <a href="http://dx.doi.org/10.15421/432533">http://dx.doi.org/10.15421/432533</a> URL: <a href="https://actualproblems.dp.ua/index.php/AP">https://actualproblems.dp.ua/index.php/AP</a></p> <p><a href="#">АІТ/article/view/322</a> (фахове видання, категорія «Б» 115)</p> <p>3. Білецький А. С., Божуха Л. М. Про граматику формальної мови гнучкого надання ресурсів веб-служб у мікросервісній архітектурі. <i>Актуальні проблеми автоматизації та інформаційних технологій</i>. 2022, no. 26. Р. 9–17. DOI: <a href="http://dx.doi.org/10.15421/432202">http://dx.doi.org/10.15421/432202</a> URL: <a href="https://actualproblems.dp.ua/index.php/AP">https://actualproblems.dp.ua/index.php/AP</a></p> <p><a href="#">АІТ/article/view/210</a> (фахове видання, категорія «Б» 115)</p>

Усі кандидатури членів ради відповідають вимогам п.п. 14, 15 «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 р. № 44 (із змінами).

**Результати голосування:**

«За» – 43 особи,  
«Проти» – немає,  
«Утримались» – немає.

**Голова  
наукового семінару**



Олена КІСЕЛЬОВА

**Секретар**



Олександр КУЗЕНКОВ