

АНАЛІЗ МОВИ ПРОГРАМУВАННЯ KOTLIN

Вінницький національний технічний університет

Анотація

Виконано аналіз мови програмування Kotlin як сучасного інструменту розробки програмного забезпечення. Розглянуто історію створення мови, її архітектурні принципи, ключові функціональні можливості, механізми забезпечення безпеки програмного коду, підтримку функціонального та об'єктно-орієнтованого програмування, особливості міжплатформної розробки та асинхронного програмування

Ключові слова: *Kotlin, мова програмування, програмна інженерія, об'єктно-орієнтоване програмування, корутини, типізація, міжплатформність.*

Abstract

The article provides a comprehensive analysis of the Kotlin programming language as a modern software development tool. The history of the language's creation, its architectural principles, key functional capabilities, mechanisms for ensuring program code security, support for functional and object-oriented programming, features of cross-platform development and asynchronous programming are considered. A comparative analysis of Kotlin with the Java language is conducted, the main advantages of using Kotlin in mobile, server-side and web development are identified. The prospects for the development of the language in the context of modern trends in software engineering are outlined.

Keywords: *Kotlin, programming language, software engineering, JVM, Android development, object-oriented programming, functional programming, coroutines, typing, cross-platform.*

Вступ

Сучасний розвиток інформаційних технологій супроводжується значним розширенням програмних систем та постійним зростанням вимог до якості програмного забезпечення. У зв'язку з цим зростає необхідність використання мов програмування, які забезпечують високі стандарти безпеки, ефективності, продуктивності та зручності у супроводі коду. Традиційні мови програмування, наприклад Java, визнані основою сучасної програмної інженерії, але мають певні недоліки: громіздкий синтаксис, складність обробки null-значень, а також збільшена ймовірність помилок під час виконання.

Ці обмеження сприяли появі нових мов, спрямованих на підвищення ефективності розробки. Однією з таких мов є Kotlin [1-5] - сучасна статично типізована мова, розроблена компанією JetBrains у 2011 році.

Суттєвий поштовх до її поширення відбувся у 2017 році, коли Google оголосив Kotlin пріоритетною мовою для розробки додатків на платформі Android. Це суттєво вплинуло на її прийняття у професійних середовищах.

Метою роботи є аналіз архітектурних особливостей мови Kotlin, вивчення її базових характеристик та переваг використання, а також визначення її місця в сучасній програмній інженерії.

Результати дослідження

Основною передумовою створення Kotlin полягала необхідність отримати мову, яка усувала б недоліки Java, зберігаючи при цьому її сильні сторони. Розробники ставили за мету створити інструмент, що підвищує безпеку коду, знижує кількість виконуваних помилок, має більш лаконічний синтаксис, підтримує сучасні теорії програмування й підвищує продуктивність розробки.

Основні етапи [1] розвитку Kotlin: перша офіційна презентація мови; 2016 - випуск стабільної версії; 2017 - офіційне визнання та підтримка платформи Android; 2019-2026 - інтенсивний розвиток Kotlin Multiplatform та інструментів для асинхронного програмування.

На сьогодні Kotlin розглядається як одна з перспективніших мов для мобільної та міжплатформної розробки. Kotlin застосовує статичну типізацію, що дозволяє виявляти помилки ще під час компіляції. Система типів цієї мови відзначається оперативним контролем, підтримкою автоматичного виведення типів і чітким розділенням nullable та non-nullable типів, що підвищує безпеку під час виконання. Механізм автоматичного виведення зменшує обсяг коду, зберігаючи його читабельність і надійність.

Kotlin реалізує концепцію null-безпеки, яка розрізняє типи, що можуть містити null, надає оператори для безпечного доступу і виконує відповідні перевірки вже на етапі компіляції. Така стратегія значною мірою підвищує стабільність програмного забезпечення.

Мова підтримує принципи об'єктно-орієнтованого програмування [3]: інкапсуляцію, наслідування, поліморфізм та абстракцію. За замовчуванням класи у мові оголошуються як final, що запобігає небажаному наслідуванню і сприяє більшій стійкості системи.

Kotlin надає можливість додавати методи та властивості до вже існуючих класів без внесення змін у їхній вихідний код, що підвищує повторне використання компонентів та адаптивність архітектури застосунків.

Data-класи автоматично формують стандартні методи для порівняння об'єктів, створення копій, перетворення в рядкове представлення та доступу до полів. Це значно полегшує роботу з моделями даних і скорочує кількість шаблонного коду.

Асинхронні функції слугують основним засобом у Kotlin для організації асинхронних обчислень. Вони дозволяють виконувати тривалі операції без блокування потоків, оптимізують використання ресурсів системи та спрощують реалізацію паралельних процесів. Застосування корутин особливо доречно в мобільній та серверній розробці.

Технологія Kotlin Multiplatform дозволяє використовувати спільну кодову базу для різних платформ, зокрема Android, веб-застосунків, серверних рішень і нативних програм. Це знижує часові та ресурсні витрати на розробку.

Kotlin Native [3] надає можливість компіляції у машинний код без проміжної віртуальної машини, відкриваючи додаткові можливості для створення високопродуктивних застосунків.

Проведемо Порівняльний аналіз Kotlin і Java.

До основних переваг Kotlin належать більш компактний синтаксис, зменшений обсяг коду, вбудована підтримка null-безпеки, можливість застосування функціональних підходів і спрощена реалізація асинхронних процесів.

Kotlin повністю сумісний з Java, що дає змогу використовувати наявні бібліотеки, поступово переходити на нову мову і інтегрувати Kotlin у поточні проекти.

Цю мову застосовують для розробки онлайн-сервісів, програм для бізнесу, складних програмних систем і сервісів, що дозволяють різним програмам обмінюватися даними між собою. Веб-розробка. Мова використовується у великих програмних комплексах завдяки високому рівню надійності та здатності до масштабування.

Впровадження Kotlin підвищує ефективність розробки, знижує кількість помилок, покращує спрощує архітектуру систем і зменшує витрати на розробку.

Незважаючи на численні переваги, Kotlin має певні обмеження: складніший процес компіляції, менша кількість навчальних матеріалів у порівнянні з Java, підвищені вимоги до ресурсів під час компіляції та відносно коротка історія застосування. Більшість цих недоліків поступово усуваються в процесі розвитку мови.

Подальший розвиток Kotlin пов'язаний із розширенням міжплатформної функціональності, оптимізацією продуктивності, удосконаленням засобів для асинхронного програмування та інтеграцією з новими технологічними рішеннями. Зростання популярності мови свідчить про її значний потенціал у майбутньому.

Kotlin поступово закріплюється як важливий інструмент сучасної програмної інженерії завдяки високій надійності, зручності використання, ефективності розробки та підтримці сучасних парадигм програмування. Мова відповідає актуальним вимогам щодо створення складних програмних систем.

Висновки

Kotlin є сучасною ефективною мовою програмування, яка забезпечує високу продуктивність розробки, безпеку виконання програм та зручність підтримки програмного забезпечення. Поєднання об'єктно-орієнтованого та функціонального програмування, підтримка міжплатформної розробки, механізми null-безпеки та сумісність із Java роблять Kotlin перспективним інструментом програмної інженерії. Подальший розвиток мови сприятиме її широкому використанню у різних сферах інформаційних технологій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Jemerov D., Isakova S. Kotlin in Action. - Manning Publications, 2017. - 360 с. - С. 1-80, 120-200.
2. Blandy J., Orendorff J., Tindall L. Programming Kotlin. - O'Reilly Media, 2019. - 450 с. - С. 40-150, 210-300.
3. Nakov S. Kotlin Programming Fundamentals. - SoftUni Foundation, 2018. - 400 с. - С. 10-120.
4. Kotlin Language Documentation. - JetBrains, 2024. - С. 1-200.
5. Android Developers Documentation: Kotlin Support. - Google Developers, 2024. - С. 5-120.

Вертепна Дарія Віталіївна - студентка групи 6ПІ-25б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, email: vertepnadasha@gmail.com.

Барчук Наталія Євгеніївна – асистент кафедри програмного забезпечення.

Романюк Олександр Никифорович – д.т.н., професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця.

Vertepna Daria Vitaliivna - student of group 6PI-25b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: vertepnadasha@gmail.com.

Barchuk Natalia Yevheniivna - assistant Professor of the Department of Software.

Romanyuk Oleksandr Nikiforovich - Doctor of Technical Sciences, Professor of the Department of Software, Vinnytsia National Technical University, Vinnytsia.