

ОГЛЯД СЕРЕДОВИЩ ТА МОВ ПРОГРАМУВАННЯ ДЛЯ НАВЧАННЯ МОБІЛЬНОЇ РОЗРОБКИ

OVERVIEW OF DEVELOPMENT PLATFORMS AND PROGRAMMING LANGUAGES FOR LEARNING MOBILE SOFTWARE DEVELOPMENT

Стаття присвячена аналізу сучасних середовищ розробки та мов програмування, які використовуються для навчання мобільної розробки. Зважаючи на постійний розвиток мобільних технологій, навчання студентів ефективної розробки мобільних застосунків стає важливим аспектом підготовки фахівців у сфері ІТ. Особлива увага приділяється порівнянню платформ для розробки Android та iOS, що дозволяє студентам оволодіти як відкритими, так і закритими екосистемами. Розглянуто ключові особливості використання мов Kotlin для Android та Swift для iOS, а також те, чому мова програмування Kotlin стала більш популярним вибором для мобільної розробки.

У статті оглянуті основні середовища розробки, зокрема Android Studio, яке є офіційним середовищем для Android і включає всі необхідні інструменти для роботи з Android SDK. Android Studio пропонує користувачам широкий набір функцій, що робить його ідеальним вибором як для новачків, так і для професіоналів. Kotlin, мова програмування для Android, відзначається своєю безпечністю, сучасним синтаксисом і підтримкою кросплатформної розробки.

З іншого боку, розробка під iOS відбувається за допомогою мови Swift. Swift забезпечує високу продуктивність, ефективність і простоту у використанні, що важливо при навчанні. Однак закритість платформи від Apple обмежує доступ до неї, що може бути суттєвим недоліком для навчальних закладів, які працюють із обмеженими матеріальними ресурсами. Swift швидко розвивається, надаючи широкий спектр можливостей для створення застосунків. Але розробка програмних засобів на iOS вимагає використання техніки Apple, що створює велику проблему для студентів.

Також у статті розглядаються альтернативні середовища розробки, такі як IntelliJ IDEA, яке дозволяє працювати не тільки з Android, але й з іншими платформами, а також Eclipse.

Таким чином, Kotlin та Android Studio є найкращим вибором для навчання мобільної розробки завдяки своїй доступності, потужності та інтеграції з сучасними інструментами, поді як Swift та його екосистема підходять більше для вузькопрофільних розробників. Оволодіння обома платформами дозволяє студентам отримати багатосторонні знання, що сприятиме їх професійному розвитку.

Ключові слова: навчання мобільної розробки, мови програмування, середовища розробки, освітні технології.

The article is devoted to the analysis of modern development environments and programming languages used to teach mobile development. Given the constant development of mobile technologies, teaching students how to develop mobile applications effectively is becoming an important aspect of IT training. Particular attention is paid to the comparison of Android and iOS development platforms, which allows students to master both open and closed ecosystems. The key features of using Kotlin for Android and Swift for iOS are considered, as well as why the Kotlin programming language has become a more popular choice for mobile development.

The article reviews the main development environments, in particular Android Studio, which is the official environment for Android and includes all the necessary tools for working with the Android SDK. Android Studio offers users a wide range of features, making it an ideal choice for both beginners and professionals. Kotlin, the programming language for Android, is noted for its security, modern syntax and support for cross-platform development.

On the other hand, iOS development is done using the Swift language. Swift provides high performance, efficiency, and ease of use, which is important in education. However, the closed nature of Apple's platform limits access to it, which can be a significant disadvantage for educational institutions working with limited material resources. Swift is developing rapidly, providing a wide range of opportunities for creating applications. However, iOS software development requires the use of Apple hardware, which poses a big problem for students.

The article also discusses alternative development environments, such as IntelliJ IDEA, which allows you to work not only with Android but also with other platforms, as well as Eclipse. Thus, Kotlin and Android Studio are the best choice for learning mobile development due to their accessibility, power, and integration with modern tools, while Swift and its ecosystem are more suitable for narrowly focused developers. Mastery of both platforms allows students to gain multifaceted knowledge that will contribute to their professional development.

Key words: mobile development education, programming languages, development environments, educational technologies.

УДК [378.091.212:004]:004.43
DOI <https://doi.org/10.32782/2663-6085/2024/75.59>

Шиян І.О.,

аспірант кафедри інформатики та кібернетики
Мелітопольського державного педагогічного університету імені Богдана Хмельницького, м. Запоріжжя

Крашенінник І.О.,

докт. філософії,
доцент кафедри інформатики та кібернетики
Мелітопольського державного педагогічного університету імені Богдана Хмельницького, м. Запоріжжя

Постановка проблеми. Мобільні технології швидко стають однією з ключових галузей програмного забезпечення, що створює високий попит на розробників мобільних застосунків. Однак, навчальні програми університетів потребують адаптації для того, щоб забезпечити студентів не лише теоретичними знаннями, але й практичними навичками з використання сучасних мов

програмування та інструментів для розробки. Постає питання вибору оптимального середовища розробки та мов програмування для навчання, які поєднують зручність, відкритість, сучасні технології та можливість швидкого освоєння. Це підтверджує **аналіз останніх досліджень і публікацій**. На сьогодні існує багато платформ і мов програмування для мобільної розробки. У дослідженні

Дамінго Леві та ін. (2021) [3] оцінюють переваги Kotlin для мобільної розробки порівняно з традиційними мовами, як Java. Автори стверджують, що Kotlin забезпечує більш лаконічний код і кращу інтеграцію з Android. Крім того, розробка Android застосунку на Kotlin надає розробнику низку додаткових синтаксичних переваг, які покращують досвід програмування та скорочують довжину коду порівняно з Java до мінімуму. У дослідженні Ардіто Луки та ін. (2020), що описує ефективність між Java та Kotlin, результати показали, що Kotlin є кращим з точки зору ефективності та узгодженості [2]. Він пропонує простоту, легкість в обслуговуванні та високу сумісність з Java. Також важливо згадати про iOS, як платформу, що пропонує власну екосистему для мобільної розробки. Згідно з дослідженням Спрінт Джини (2019) [6], середовище Xcode, яке використовується для розробки під iOS, є дуже потужним інструментом з точки зору інтеграції з апаратними засобами. Однак, закритість платформи і високі вимоги до використання продукції Apple роблять її менш привабливою для освітніх програм, особливо через високі витрати на апаратні ресурси.

Попри численні наукові дослідження у галузі мобільної розробки, залишається відкритим питання щодо оптимального середовища та мови програмування для навчання студентів. Чи варто одночасно навчати студентів розробці на Android і iOS, чи краще зосередитися на одній платформі для поглибленого вивчення? Як краще організувати навчальний процес, щоб студенти отримали знання, які допоможуть їм бути конкурентоспроможними на ринку праці? Це дає підстави стверджувати про те, що означений вище аспект ми можемо розглядати **як виділення невирішених раніше частин загальної проблеми дослідження.**

Мета полягає в тому, щоб проаналізувати популярні середовища і мови програмування для мобільної розробки і надати рекомендації щодо оптимального вибору і використання інструментів для навчання мобільній розробці, орієнтуючись на доступність, сучасність та перспективність платформ і мов програмування.

Виклад основного матеріалу дослідження. Сучасна мобільна розробка є важливою частиною навчальних програм у багатьох університетах по всьому світу. Вибір між платформами Android та iOS під час навчання студентів має великий вплив на подальшу кар'єру розробників. Таким чином, важливо порівняти ці дві основні мобільні операційні системи, зосереджуючись на інструментах для розробки, доступності ресурсів і кар'єрних перспективах, щоб з'ясувати, яка з них краще підходить для навчальних програм.

Одним із ключових аспектів під час вибору платформи для розробки мобільних застосунків

є вибір мови програмування, яку студенти будуть вивчати. Для Android розробки традиційно використовуються Java та Kotlin, а для iOS – Swift. Розглянемо детальніше ці мови, їхні переваги та те, чому Kotlin вважається найкращим вибором для навчання студентів.

1. **Swift.** Swift – це мова програмування від Apple для розробки застосунків на платформах iOS, macOS, watchOS та tvOS. Swift був представлений у 2014 році як заміна Objective-C і має кілька ключових переваг для новачків у програмуванні [7]:

- **Читабельний синтаксис.** Swift спроектований таким чином, щоб бути простішим і зрозумілішим порівняно з Objective-C. Це дозволяє студентам швидше освоїти основи розробки, оскільки синтаксис Swift є ближчим до мов високого рівня.

- **Безпека.** Swift має вбудовані механізми для запобігання помилок, таких як обробка null значень та типова безпека. Це допомагає студентам уникати поширених помилок під час навчання програмуванню.

- **Підтримка найсучасніших технологій.** Swift постійно розвивається та отримує підтримку нових технологій Apple, таких як SwiftUI.

- **Однак, незважаючи на ці переваги, Swift має кілька обмежень у контексті навчання:**

- **Обмеженість платформою.** Swift використовується лише для розробки під екосистему Apple. Це означає, що студенти, які вивчать Swift, будуть обмежені платформою iOS/macOS/tvOS тощо. Це зменшує поле застосування навичок, які студенти можуть отримати, порівняно з мовами, що використовуються на абсолютно різних платформах.

- **Доступність інструментів.** Для роботи з Swift потрібне середовище Xcode, яке працює лише на macOS. Це може створити фінансові обмеження для навчальних закладів і студентів, які не мають доступу до техніки Apple.

2. **Java/Kotlin.** Java є однією з найбільш популярних мов програмування. Вона використовується для Android-розробки з моменту запуску платформи, а також є основною мовою у багатьох інших галузях IT, таких як серверна розробка, веб-розробка і навіть у розробці великих корпоративних систем. Переваги Java включають:

- **Кросплатформність.** Java є мовою загального призначення, що дозволяє студентам використовувати її не тільки для Android, а й для веб-розробки, розробки серверних застосунків тощо.

- **Широка спільнота спеціалістів та навчальних ресурсів.** Java існує вже понад 25 років, тому має величезну кількість матеріалів, бібліотек і інструментів, доступних для студентів.

- **Однак Java має певні недоліки у контексті мобільної розробки:**

- **Застарілий синтаксис.** Синтаксис Java може здатися студентам складним та менш зручним

у порівнянні з більш сучасними мовами, такими як Swift чи Kotlin.

- Повільність розвитку. Хоча Java залишається популярною, вона не завжди швидко адаптується до сучасних вимог мобільної розробки, що робить її менш привабливою для створення застосунків.

- Kotlin – це сучасна мова програмування, що стала офіційною мовою Google для розробки Android-застосунків з 2017 року. Вона стала популярним вибором серед студентів і професійних розробників завдяки своїм численним перевагам, серед яких [4]:

- Сучасний та зручний синтаксис. Kotlin був розроблений для того, щоб виправити обмеження Java і зробити розробку простішою та більш ефективною. Його синтаксис є лаконічним, зрозумілим і ближчим до Swift або Python. Наприклад, в Kotlin існує багато операцій, що потребують декількох рядків коду в Java, можуть бути виконані в одну команду. Це спрощує навчання і скорочує кількість помилок, які можуть допустити студенти.

- Повна сумісність з Java. Kotlin повністю сумісний з Java, що дозволяє студентам використовувати обидві мови в одному проекті. Це забезпечує плавний перехід від старих проектів на Java до нових на Kotlin та зберігає всі переваги багаторічного використання Java.

- Безпека і запобігання помилок. Kotlin має вбудовані функції для зниження кількості типових помилок. Наприклад, Kotlin запобігає NullPointerException (одна з найпоширеніших проблем у Java) через вбудовану систему обробки nullable типів.

- Кросплатформність. Хоча Kotlin спочатку розроблявся для Android, зараз він також підтримує розробку під різні платформи (Kotlin Multiplatform). Це робить Kotlin більш привабливим для навчання у порівнянні зі Swift, який орієнтований тільки на екосистему Apple.

3. Порівняння Swift та Kotlin:

- Читабельність і зручність. Обидві мови – Swift і Kotlin – мають сучасний синтаксис, який дозволяє створювати чистий і лаконічний код. Однак Kotlin надає більше можливостей для студентів завдяки своїй сумісності з Java та можливості використовувати її за межами мобільної розробки.

- Кросплатформність. Kotlin забезпечує більше кар'єрних можливостей, оскільки його можна використовувати для розробки Android-застосунків, веб-застосунків, серверних систем та навіть iOS-застосунків через Kotlin Multiplatform [5]. Swift, у свою чергу, залишається прив'язаним до екосистеми Apple.

- Доступність інструментів. Kotlin і Android Studio доступні безкоштовно і можуть використовуватися на різних операційних системах, тоді як Swift вимагає використання Xcode на macOS,

що може обмежити доступ студентів до цієї мови через вартість обладнання.

Обираючи між Swift та Java/Kotlin для навчання студентів, важливо враховувати такі аспекти:

Kotlin є більш універсальною мовою програмування, яка забезпечує ширші кар'єрні можливості завдяки сумісності з Java та можливості розробки для кількох платформ. Сучасний синтаксис і безпечні механізми Kotlin роблять її зручною для навчання новачків.

Swift є привабливим варіантом для студентів, які хочуть спеціалізуватися на екосистемі Apple. Його простота і безпека також роблять його гарним вибором для початківців, але обмеженість платформою Apple може стати перешкодою. Відсутність матеріального забезпечення університетів технікою Apple робить Swift та iOS недоступним наразі варіантом для навчання студентів.

З урахуванням усіх факторів, Android та Kotlin виглядає більш привабливим вибором для українських університетських навчальних програм.

Для розробки на платформу Android існує кілька середовищ розробки, які використовуються як професіоналами, так і студентами. Кожне середовище має свої переваги та недоліки. Найбільш популярні середовища – це Android Studio, IntelliJ IDEA та Eclipse.

1. **Android Studio.** Android Studio – офіційне середовище розробки для Android, яке надається Google і базується на IntelliJ IDEA. Воно стало основним інструментом для розробників завдяки своїм численним перевагам [1]:

- Офіційна підтримка: Android Studio є офіційним IDE для розробки Android-додатків, тому воно надає повний доступ до всіх нових API та функцій Android відразу після їхнього випуску.

- Інтеграція з інструментами Android: IDE підтримує всі важливі інструменти для розробки, такі як Android SDK, Android Virtual Device (AVD) для тестування додатків, Gradle для автоматизації збірок, а також інтеграцію з Firebase для додавання серверних функцій.

- Зручність для студентів та новачків: Android Studio надає зручний інтерфейс і потужні інструменти для відлагодження, аналізу продуктивності та роботи з ресурсами UI, що робить його ідеальним вибором для навчання новачків.

- Підтримка Jetpack Compose: Android Studio має відмінну підтримку Jetpack Compose – сучасного фреймворка для створення UI на Android, що значно спрощує розробку користувацького інтерфейсу.

- Головним недоліком Android Studio є високі вимоги до комп'ютерних ресурсів, що може суттєво вплинути на продуктивність, особливо на старіших або менш потужних машинах.

2. **IntelliJ IDEA.** IntelliJ IDEA – це універсальне середовище розробки від компанії JetBrains, яке

також використовується для розробки Android-застосунків. Android Studio базується на IntelliJ IDEA, тому в нього є схожі переваги:

- Універсальність. IntelliJ IDEA підтримує розробку не тільки для Android, але й для інших платформ, таких як веб, серверні рішення та інші мобільні платформи.

- Широкий функціонал. IDEA пропонує безліч інструментів для відлагодження, рефакторингу та аналізу коду, що робить його корисним для досвідчених розробників, які працюють над великими проектами.

- Недоліки IntelliJ IDEA:

- Менш специфічні інструменти для Android. IntelliJ IDEA має підтримку Android, але спеціалізовані інструменти для цієї платформи краще інтегровані в Android Studio.

- Вартість. Хоча є безкоштовна версія IntelliJ IDEA Community Edition, повна версія з розширеними можливостями платна, що може бути важливо для студентів і навчальних закладів.

3. **Eclipse.** Eclipse – одне з найстаріших середовищ для розробки Java-програм і мобільних застосунків для Android. Хоча воно не є офіційним середовищем для Android і його популярність значно знизилася після виходу Android Studio, деякі розробники все ще використовують його:

- Кросплатформеність: Eclipse може використовуватися на різних операційних системах, і, як універсальне IDE, підтримує не тільки Java, але й безліч інших мов.

- Плагіни для Android: Раніше Eclipse використовувалося разом з Android Development Tools (ADT), що дозволяє створювати Android-застосунки, але з часом Google припинила офіційну підтримку ADT на користь Android Studio.

- Це середовище розробки також має такі недоліки:

- Застарілий підхід: Після випуску Android Studio Eclipse більше не отримує офіційної підтримки для Android-розробки, що робить його менш привабливим варіантом для студентів і новачків.

- Обмежена функціональність: Порівняно з Android Studio, Eclipse має меншу функціональність для роботи з сучасними Android-технологіями, такими як Jetpack Compose.

Отже, **Android Studio** є найкращим вибором для розробки Android-застосунків завдяки офіційній підтримці Google, інтеграції з усіма інструментами для Android і підтримці сучасних технологій. Важливим фактором є те, що Android Studio поширюється безкоштовно, тому студенти можуть використовувати Android SDK без додаткових витрат.

Висновки. З огляду на проведений аналіз, можна зробити висновок, що мобільна платформа Android, середовище розробки Android Studio та мова програмування Kotlin є найбільш оптимальними інструментами для навчання студентів мобільній розробці. Перевагами Android є її відкритість, доступність для різних операційних систем та підтримка великої кількості пристроїв, що робить її зручнішою для навчальних програм. Kotlin є сучасною мовою програмування, яка має спрощений синтаксис та зручні інструменти для розробки додатків, що робить її пріоритетним вибором при навчанні студентів мобільної розробки.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Android Studio & App Tools – Android Developers. *Android Developers*. URL: <https://developer.android.com/studio> (дата звернення: 10.09.2024).
2. Effectiveness of Kotlin vs. Java in android app development tasks / L. Ardito et al. *Information and Software Technology*. 2020. Vol. 127. P. 106374. URL: <https://doi.org/10.1016/j.infsof.2020.106374> (дата звернення: 10.09.2024).
3. Damingo L., Elliot K. N. A Comparative Review of Mobile Application Development Frameworks: Kotlin Vs Java. *J. Phys. Conf. Ser.* 2022. Vol. 6, no. 3. P. 112–121.
4. Kotlin Docs | Kotlin. *Kotlin Help*. URL: <https://kotlinlang.org/docs/home.html> (дата звернення: 10.09.2024).
5. Kotlin Multiplatform | Kotlin. *Kotlin Help*. URL: <https://kotlinlang.org/docs/multiplatform.html> (дата звернення: 10.09.2024).
6. Sprint G. Mobile App Development. *SIGCSE '19: The 50th ACM Technical Symposium on Computer Science Education*, Minneapolis MN USA. New York, NY, USA, 2019. URL: <https://doi.org/10.1145/3287324.3293822> (дата звернення: 10.09.2024).
7. Swift – Apple Developer. *Apple Developer*. URL: <https://developer.apple.com/swift> (дата звернення: 10.09.2024).