

РОЗДІЛ 4. ОПТИМІЗАЦІЯ НАВЧАННЯ ДИСЦИПЛІН ПРИРОДНИЧО-МАТЕМАТИЧНОГО ЦИКЛУ ЗАСОБАМИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004: 378

DOI10.5281/zenodo.12190915

В. М. Базурін

ORCID ID 0000-0002-6614-4889

Державний торговельно-економічний університет

ОСОБЛИВОСТІ НАВЧАННЯ МАЙБУТНІХ БАКАЛАВРІВ КОМП'ЮТЕРНИХ НАУК ОСНОВ ПРОГРАМУВАННЯ НА МОВІ KOTLIN

У статті розглянуто проблеми навчання студентів комп'ютерних спеціальностей програмуванню на мові Kotlin. Вивчення мови Kotlin зазвичай викликає у студентів певні проблеми, пов'язані з тим, що синтаксис мови Kotlin відрізняється від синтаксису мов, які спираються на C і C++. Як доводять вчені, мову Kotlin доцільно вивчати після вивчення основ програмування. У процесі навчання розробки мобільних додатків на мові Kotlin у студентів виникають труднощі, пов'язані з розумінням структури проекту мобільного додатка для Android і пов'язані з відмінністю синтаксису мови Kotlin від синтаксису мов C, C+, Java, C# та інших.

Саме тому актуальними питаннями навчання розробки мобільних додатків на Kotlin є комбінація задачного та проектного підходів до вивчення програмування на Kotlin, завдяки чому студенти спочатку поетапно виконують порівняно прості завдання з вивчення основних алгоритмічних конструкцій, основ об'єктно-орієнтованого програмування, колекцій. Лише після цього доцільно переходити до вивчення структури додатка Android, створення мобільних додатків в середовищі Android Studio.

У процесі навчання програмування на мові Kotlin студенти використовують різні середовища. Найбільш поширеним середовищем розробки мобільних додатків для операційної системи Android є Android Studio. Проте через високі системні вимоги дане середовище розробки може бути встановлене не на всіх комп'ютерах. Тому на початковому етапі вивчення програмування на Kotlin доцільно використати онлайн середовище Kotlin Playground або офлайн середовище IntelliJ IDEA.

***Ключові слова:** освітній процес у закладі вищої освіти, задачний підхід, комп'ютерні науки, програмування, мобільний додаток, Kotlin, студенти.*

Постановка проблеми. Інформаційні технології увійшли практично в усі сфери життя людини. Вони постійно розвиваються і оновлюються. Багато важливих технологій пов'язані з мобільними додатками (наприклад, Дія). Постійний розвиток таких технологій потребує наявності відповідних фахівців, які створюватимуть мобільні додатки. Підготовку таких фахівців зазвичай здійснюють заклади вищої освіти, але в загальноосвітній школі, під час вивчення алгебри, геометрії, інформатики, фізики та хімії формуються такі кості майбутніх програмістів, як алгоритмічне та аналітичне мислення, а в процесі вивчення інформатики учні можуть на практиці перевірити, чи подобається їм створювати програми.

З іншого боку, створення мобільних додатків викликає певні труднощі. Наприклад, для створення додатків для операційної системи Android використовується одна з двох мов (Java або Kotlin), а необхідним компонентом вивчення є структура додатка Android. Мова Java має C-подібний синтаксис, але останнім часом для створення мобільних додатків для Android все частіше використовується мова Kotlin, яка має значні відмінності від C і C++ з точки зору синтаксису.

Саме тому метою статті є дослідження методики навчання програмування на мові Kotlin.

Аналіз актуальних досліджень. Мова Kotlin є однією з об'єктно-орієнтованих мов програмування, тому її закономірності її вивчення такі ж самі, як і інших мов програмування. Водночас синтаксис і логіка програм на мові Kotlin (особливо Android-додатків) мають певні відмінності.

Дослідження, пов'язані з використанням Kotlin для розроблення додатків, можна умовно розбити на такі групи: 1) дослідження всіх технічних та алгоритмічних питань, пов'язаних з використанням мови Kotlin для розробки додатків для Android; 2) визначення особливостей навчання програмування на мові Kotlin (навчання розробці додатків для Android на мові Kotlin).

До досліджень першої групи віднесемо наступні. М. Martinez і В. Gois Mateus здійснили емпіричне дослідження якості додатків Android, написаних на мові Kotlin. У результаті дослідження з'ясовано, що на мові Kotlin написано лише 11,26% додатків з кількості додатків, які знаходяться у відкритому доступі [2].

Інші науковці досліджували застосування методів глибинного навчання під час перенесення проектів з мови Java на мову Kotlin [3].

Предметом досліджень L.Ardito, R.Coppola, G.Malnati, M.Torchiano є ефективність додатків, написаних на мові Kotlin, і порівняння їх ефективності з додатками, написаними на мові Java [3].

М. Martinez і В. G. Mateus досліджували проблему міграції додатків для Android на мову Kotlin. Основними причинами міграції додатків на Kotlin, на думку авторів, є: уникнення помилок, використовуючи безпечніший код; слідувати за рекомендаціями Google; з метою навчання; використання мови, на 100% сумісної з мовою Java; використання функціональної мови програмування для розробки додатків для Android; уникання застосування мови Java; уникання фрагментації платформи Android і обмежень версій Java; досягнення мультиплатформенної розробки; використання можливостей інших мов програмування [5].

Проблему навчання програмування на мові Kotlin досліджували Р. Spath [6], N. Dimitrijević, V. Milicevic, D. Cvijanovic, N. Zdravković досліджували результати застосування автоматизованої системи навчання програмування на Kotlin. Дослідники зазначають, що у студентів виникають значні труднощі під час вивчення Kotlin у якості першої мови програмування, тому доцільно вивчати Kotlin в якості другої або третьої мови програмування, після вивчення Java [8, с.139]. дослідники зазначають, що Kotlin має широкі перспективи для застосування його під час створення мобільних додатків під Android. Відповідно на ринку праці існуватиме попит на програмістів, які володіють Kotlin.

В іншій своїй праці [9] ці дослідники обґрунтовують структуру та інтерфейс автоматизованої системи перевірки програм, написаних на Kotlin. У статті також наведено результати екзамену для Android-розробників, які вивчали Kotlin з застосуванням автоматизованої системи перевірки програм. На думку авторів найбільш ефективним є застосування не тільки автоматизованої системи перевірки програм, а створення повністю автоматизованої системи, яка повністю буде забезпечувати вивчення студентами курсу програмування на Kotlin [9].

Є.О. Беркунський, А.Ю. Павленко та І.Л. Михелев відмічають такі переваги мови Kotlin над іншими мовами: простота і читабельність коду; 100% сумісність з Java; безпека коду; низка розширень, які надають можливість зробити код більш компактним [11].

Я.В. Мосляков, А.О. Овчарова і Л.П. Остапенко зазначили, що значна кількість учнів виявили інтерес до створення мобільних додатків. У цій же статті дослідники описують варіанти вивчення різних технологій і мов створення мобільних додатків: MIT App Inventor, Python, Kivu [12].

Отже, проблема навчання програмуванню на мові Kotlin у науково-методичній літературі розкрита недостатньо.

Метою статті було дослідити проблему навчання студентів комп'ютерних спеціальностей програмуванню на мові Kotlin.

Виклад основного матеріалу. Мова програмування Kotlin є однією з двох мов, які переважно використовуються під час розроблення мобільних додатків для Android.

Мову Kotlin було розроблено фірмою JetBrains в 2011 році для роботи з платформою Java. Відповідно, Kotlin повністю сумісний з Java, додаток на Kotlin зазвичай компілюється в байт-код, який може виконуватися на віртуальній машині Java (JVM). Саме завдяки такій особливості Kotlin розробники програмного забезпечення можуть переходити на Kotlin, не переписуючи всю систему наново.

Мова Kotlin застосовує статичну типізацію. Тобто, тип змінної визначається під час її оголошення і не може бути зміненим під час роботи програми.

Kotlin підтримує функціональне програмування. Мова Kotlin надає програмісту значну кількість інструментів і конструкцій. Завдяки функціональному програмуванню код на Kotlin стає більш компактним. Мова Kotlin також має низку інструментів, які запобігають помилкам на етапі компіляції.

Основними сферами застосування мови Kotlin є:

- розроблення мобільних додатків для Android;
- розроблення серверних додатків;
- розроблення додатків для Desktop комп'ютерів;
- WEB-розробка;
- загальні задачі програмування.

У наш час мова Kotlin підтримується компанією Google, і система розробки Android Studio підтримує її.

Основними перевагами Kotlin, на думку [1], є:

- збільшення продуктивності додатків;
- збільшення рівня безпеки;
- більш зрозумілий і читабельний код;
- значна екосистема бібліотек та інструментів.

Поряд з перевагами Kotlin має і низку недоліків:

- необхідність переходу з Java на Kotlin;
- обмежене застосування в деяких галузях (системне програмування, розроблення ігор тощо);
- інший синтаксис у порівнянні з мовами, які базуються на C і C++ (Java, C# тощо) [1].

Питання вивчення мови програмування Kotlin невідривно пов'язані з потребою в інтегрованому середовищі розроблення (IDE). На даний момент, на нашу думку, найбільшої уваги заслуговують Android Studio і JetBrains IntelliJ IDEA. Існують також онлайн-середовища, але їх можливості значно обмежені. Розглянемо їх характеристики детальніше.

Android Studio – це середовище розроблення програм, створене фірмою Google. В цьому середовищі можна спроектувати інтерфейс будь-якого Android-додатка, написати програмний код і протестувати додаток або на мобільному телефоні, або на емуляторі мобільного телефону. Тобто, IDE Android Studio цілком достатньо, щоб створювати і тестувати мобільні додатки для Android, написані на Kotlin. На нашу думку, Android Studio має 2 дуже важливих недоліки: 1) порівняно високі системні вимоги (на момент написання статті поточна версія Android Studio потребувала 16 ГБ оперативної пам'яті); 2) необхідність встановлювати на комп'ютері найновішу операційну систему і найновішу версію Android Studio. Саме через це застосовувати Android Studio для вивчення програмування на Kotlin можуть не всі учні і навіть не всі студенти. Високі системні вимоги звужують навчальні можливості цього середовища програмування, а для початківця досить важким елементом є вивчення структури проекту додатка для Android.

Інше середовище – IntelliJ IDEA – має дві версії: Common і Ultimate [13].

Під час роботи з IntelliJ IDEA потрібно створювати проект додатка, але його структура значно простіша і зрозуміліша, ніж структура проекту додатка в Android Studio. На нашу думку, для вивчення основ програмування на Kotlin більше підходить IntelliJ IDEA.

Слід зазначити також, що існують і онлайн-середовища. Найбільш часто використовуване online IDE Kotlin Playground [14]. Для програміста-початківця Kotlin Playground надає всі потрібні засоби та інструменти.

Отже, для вивчення основ програмування на Kotlin доцільно використати Kotlin Playground, а потім, після відпрацювання основних алгоритмічних конструкцій, основ об'єктно-орієнтованого програмування, можна переходити до роботи в IDE IntelliJ IDEA або Android Studio.

Стосовно місця Kotlin у вивченні програмування. Значна кількість науковців [6; 10] вважає, що вивчати Kotlin в якості першої мови програмування недоцільно. Для початківця синтаксичні особливості Kotlin можуть вчинити негативний вплив на подальше вивчення програмування (C/ C++, Java, C# тощо). Ми погоджуємося з такою думкою, адже маємо вже певний досвід навчання програмування на Kotlin студентів 4-го курсу спеціальності 122 «Комп'ютерні науки», для яких Kotlin вже не був першою мовою програмування.

Слід відзначити, що в мові Kotlin оператор циклу `for` використовується для перебору колекцій (рядків, масивів, списків), а у звичному для нас вигляді – для табулювання функцій – він може бути використаний, але для цього спочатку значення лічильника помістити в масив.

Існують також інші відмінності, через які Kotlin недоцільно використовувати в якості першої мови програмування.

Стосовно навчання програмування на Kotlin, то тут, на нашу думку, існує 2 основних підходи: задачний і проектний.

Задачний підхід передбачає написання студентами на Kotlin порівняно невеликих програм різного рівня складності. Таким чином відпрацьовуються основні алгоритмічні конструкції, засвоюються особливості різних типів даних, визначаються особливості об'єктно-орієнтованого програмування.

Переваги: поступове і системне вивчення особливостей мови Kotlin.

Недоліки: великі витрати часу (адже доводиться вивчати всі алгоритмічні конструкції, типи даних, особливості ООП), порівняно незначний зв'язок з практикою. Студентам, які мають високий рівень алгоритмічного мислення, може бути просто нецікаво поступово і детально вивчати особливості мови Kotlin так, як вони раніше вивчали особливості Java.

Проектний підхід передбачає створення відразу проекту – додатка для Android за допомогою Android Studio. Всі синтаксичні особливості мови Kotlin, логіка програм та інше вивчаються студентами самостійно, у процесі розроблення додатка. Перевага такого підходу полягає в тому, що в результаті кожної лабораторної роботи студенти створюють повноцінний додаток для ОС Android. Це сприяє мотивації студентів до вивчення програмування.

Проте такий підхід має і певні недоліки:

- складність. Для деякої частини студентів важко пов'язати між собою окремі компоненти додатка Android;
- деяка безсистемність. З особливостей мови Kotlin вивчаються лише ті, які потрібні для створення додатка. Всі інші питання залишаються на розсуд студента.

На нашу думку, оптимальним шляхом розв'язання цієї проблеми є поєднання задачного і проектних підходів у процесі вивчення Kotlin. На користь такого підходу:

- всі студенти, навіть ті, які мають високий рівень алгоритмічного мислення, повинні вивчити синтаксис Kotlin хоча б частково. Вивчати основи Kotlin, як було зазначено вище, краще за допомогою задач;
- ті студенти, який мають середній рівень розвитку алгоритмічного мислення, можуть засвоїти синтаксис мови програмування Kotlin, відмінний від синтаксису C-подібних мов;
- після засвоєння основ синтаксису і семантики Kotlin студенти матимуть можливість зробити вибір між подальшим виконанням лабораторних робіт і об'ємним проектом.

На основі аналізу науково-методичних джерел нами запропоновано наступну тематику лабораторних занять. При цьому враховано, що мова Kotlin є другою або третьою мовою програмування, яку вивчають студенти.

1. Середовище розробки Kotlin Playground. Лінійні алгоритми.
2. Розгалужені алгоритми. Множинний вибір (when).
3. Перетворення типів. Функції.
4. Цикли.
5. Класи.
6. Data class. Sealed class.
7. Колекції. List.
8. Колекції. Set.
9. Колекції. Map.
10. Створення додатка з однією Activity в Android Studio.
11. Створення додатка, який зберігає дані в файлі.
12. Створення додатка з кількома Activity.

Отже, нами уточнено зміст навчання програмуванню на Kotlin. Визначимо форми, методи і засоби навчання.

Стосовно форм навчання – це мають бути лабораторна робота і самостійна робота. Під час лабораторної роботи студенти створюють програми на мові Kotlin, причому у випадку неможливості виправлення помилок вони можуть звернутися до викладача.

У процесі навчання студентів програмуванню на Kotlin провідне місце належить самостійній роботі студентів. У процесі самостійної роботи під час виконання лабораторних робіт у них формуються навички розроблення програм на мові Kotlin.

Щодо методів навчання, то одним з найбільш важливих методів є метод наочних прикладів. Для того, щоб успішно засвоїти програмування на мові Kotlin, студенти повинні отримувати приклади задач і їх розв'язків.

Висновки та перспективи подальших наукових розвідок. Отже, у студентів виникають певні труднощі під час вивчення програмування на мові Kotlin. Ці труднощі пов'язані з особливостями синтаксису Kotlin, високими системними вимогами IDE Android Studio, структурою проекту додатка для Android.

Перспективами подальших наукових пошуків є розроблення методичної системи навчання програмування на Kotlin і експериментальна перевірка її ефективності. Особливої уваги потребує змістовий компонент методичної системи: посібник з мови Kotlin, лабораторний практикум, збірник тестів. Це той компонент методичної системи, який постійно оновлюється, оскільки мова Kotlin постійно розвивається і оновлюється.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ / REFERENCES

1. Kim, M., Kim, Y., Jeong, H., Heo, J., Kim, S., Chung, H., & Lee, E. (2022). An empirical study of deep transfer learning-based program repair for kotlin projects. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1441–1452).
2. Góis, Mateus, B., Martinez, M. (2019). An empirical study on quality of Android applications written in Kotlin language. Empirical Software Engineering, 24(6), 3356–3393.
3. Ardito, L., Coppola, R., Malnati, G., Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in android app development tasks. Information and Software Technology, 127, 106374.
4. Oliveira, V., Teixeira, L., Ebert, F. (2020). On the adoption of kotlin on android development: A triangulation study. In 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 206–216). IEEE.
5. Martinez, M., Mateus, B. G. (2021). Why did developers migrate android applications from java to kotlin?. IEEE Transactions on Software Engineering, 48(11), 4521–4534.
6. Späth, P. (2019). Learn Kotlin for Android Development. Apress.

7. Syaifudin, Y. W., Funabiki, N., & Liem, I. (2021). Comparisons of Student's Self-Learning Performances Using Java and Kotlin Languages in Android Programming Learning Assistance System. In 2021 1st Conference on Online Teaching for Mobile Education (OT4ME) (pp. 93–97). IEEE.
8. Dimitrijević, N., Milicevic, V., Cvijanovic, D., Zdravković, N. (2021). Learning the Kotlin programming language using an autograding system.
9. Dimitrijevic, N., Zdravkovic, N., Milicevic, V. (2023). An Automated Grading Framework for the Mobile Development programming language Kotlin. International Journal for Quality Research.
10. Bazurin, V. M., Chashechnikova, O. S., Karpenko, Ye. M., Pursky, O. I., Palchuk, P. M. (2022). Interdisciplinary problems of mathematical content as a means of teaching programming to a secondary school pupils. Journal of Physics. Conference series, 2288. 012010 IOP Publishing doi:10.1088/1742-6596/2288/1/-120.
11. Беркунський, Є. Ю., Павленко, А. Ю., Михелев, І. Л. (2023). Сучасні засоби розробки та мови програмування у навчальному процесі. Інновації в суднобудуванні та океанотехніці : матеріали XIV Міжнародної науково-технічної конференції. Миколаїв, (сс. 415–417). (Berkunskiyi, Ye. Yu., Pavlenko, A. Yu., Mykheliev, I. L. (2023). Modern development tools and programming languages in the educational process. In Innovations in shipbuilding and ocean engineering: materials of XIV International scientific and technical conference.: Mykolaiv, (pp. 415–417).
12. Мосляков, Я. В., Овчарова, А. О., Остапенко, Л. П. (2020). Проблеми навчання учнів основ створенню мобільних додатків в сучасному шкільному курсі інформатики. Науково-дослідна робота студентів як чинник удосконалення професійної підготовки майбутнього вчителя: зб. наук. Харків, (сс.74–79). (Mosliakov, Ya. V., Ovcharova, A. O., Ostapenko, L. P. (2020). Problems of teaching students the basics of creating mobile applications in a modern school computer science course. Scientific and research work of students as a factor in improving the professional training of the future teacher: coll. of science Kharkiv, (pp.74–79).).
13. Download IntelliJ IDEA. <https://www.jetbrains.com/idea/download/?section=windows>.
14. Kotlin Playground. URL: <https://play.kotlinlang.org/>

Bazurin V. M. Features of education of future bachelors of computer sciences in the fundamentals of programming in the Kotlin language.

The article discusses the problems of teaching computer majors students to program in the Kotlin language. Learning Kotlin usually causes some problems for students due to the fact that the syntax of Kotlin is different from the syntax of languages based on C and C++. As scientists prove, it is advisable to learn the Kotlin language after learning the basics of programming. In the process of learning how to develop mobile applications in the Kotlin language, students have difficulties in understanding the structure of a mobile application project for Android and related to the difference between the syntax of the Kotlin language and the syntax of C, C+, Java, C# and others.

That is why the topical issues of teaching the development of mobile applications on Kotlin are the combination of problem-based and project-based approaches to learning programming on Kotlin, thanks to which students first step by step perform relatively simple tasks to study basic algorithmic structures, the basics of object-oriented programming, and collections. Only after that, it is advisable to move on to studying the structure of the Android application, creating mobile applications in the Android Studio environment.

In the process of learning Kotlin programming, students use different environments. The most common mobile application development environment for the Android operating system is Android Studio. However, due to high system requirements, this development environment may not be installed on all computers. Therefore, at the initial stage of learning Kotlin programming, it is advisable to use the Kotlin Playground online environment or the IntelliJ IDEA offline environment.

Key words: *educational process in a higher education institution, problem-based approach, computer scienc, programming, mobile application, Kotlin, students.*