**АНАЛІЗ ПРОБЛЕМАТИКИ ПРОГРАМНИХ ЗАСОБІВ ДО МОБІЛЬНОГО ЗАСТОСУВАННЯ З ОРГАНІЗАЦІЇ ПРОВЕДЕННЯ НАСТІЛЬНИХ ІГОР**

**ANALYSIS OF THE PROBLEMS OF SOFTWARE TOWARDS MOBILE APPLICATIONS FOR THE ORGANIZATION OF BOARD GAMES**

Науковий керівник: к.т.н., доцент кафедра інформаційних технологій

Рудніченко Микола Дмитрович

Здобувач бакалаврату Затоковенко Дар'я Геннадіївна

Supervisor: (Ph.D), assistant professor Department of Information Technology

Rudnichenko Mykola Dmytrovych

Bachelor's candidate Zatokovenko Daria Gennadievna

***Анотація:*** *розглянуто проблеми, особливості призначення,* *зроблений* *аналіз проблематики програмних засобів до мобільного застосування з організації проведення настільних ігор. Розглянуто наслідки та складності написання коду найбільш розповсюдженою мовою програмування.*

***Ключові слова****: мобільні застосування, Kotlin, Java, Android.*

***Abstract****: the problems, features of purpose were considered, an analysis of problems of software for mobile application on the organization of board game events was made. The consequences and difficulties of writing code in the most common programming language are considered.*

***Keywords****: mobile applications, Kotlin, Java, Android.*

Довгий час основною мовою программування програмного забезпечення на операційну систему Android від Google Inc залишалась Java. За роки домінування серед розробників мобільних застосунків було знайдено чимало недоліків, один з них – Java дуже громіздка для кодування.

Оскільки вона є надто потужною і крос-платформною (на пристрої, який використовується для її читання, інстальовано програму Java та середовище виконання), процес розробки стає невиправдано складнішим.

Java - мова програмування з декількома парадигмами, яка підтримує об'єктно-орієнтоване програмування.

Мова програмування Java і програмна платформа Java піддавалися критиці за вибір дизайну, включаючи реалізацію загальних засобів, примусове об'єктно-орієнтоване програмування, обробку беззнакових чисел, реалізацію арифметики з плаваючою комою та історію вразливостей безпеки в первинній Java.

Програмне забезпечення, написане на Java, особливо його ранні версії, піддавалося критиці за свою продуктивність порівняно з програмним забезпеченням, написаним іншими мовами програмування [1].

У процесі написання коду використовуються об'єкти (змінну, функцію або майже будь-який набір даних), і використовуються класи для визначення типу та початкових даних всередині об'єкта.

Розробнику потрібно обов'язково включати правильні бібліотеки, які дозволяють класам визначати об'єкт.

Котлін полегшує велику частину цих громіздких речей. Розробнику все ще потрібна Java, встановлена на пристрої, на якому запущені програми, написані на Kotlin, оскільки вона створює програму, яка працює так само, як і в тому випадку, якщо вона написана на "чистій" Java [2].

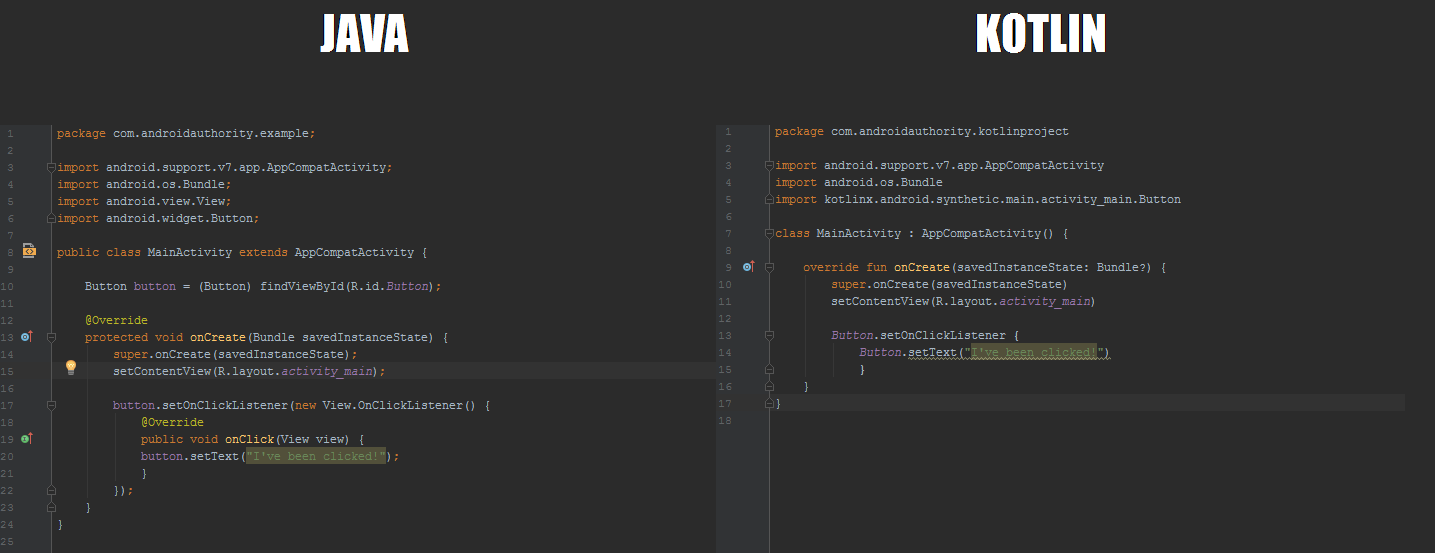


Рисунок 1 – Порівняння обсягу коду Java та Kotlin однієї функції

Проблеми Java, які вирішує Kotlin:

* Немає власної підтримки для опціональних компаній:

хоча існує клас Optional <T>, його використання передбачає генерацію великої кількості типового коду, який можна зберегти, якби підтримка параметрів була побудована в самій мові, а не як її розширення [3, 4];

* Вбудована підтримка функціонального програмування відсутня:

в Java існує Stream API (він має підтримку лише в Android, починаючи з API 24), але його інтеграція в мову схожа на необов’язкову, вона погано коіснує в об'єктах, пов'язаних з примітивними типами (IntStream, DoubleStream і т.д.), і через клас Stream <T> для всіх інших об'єктів;

* Підтримка анонімних функцій (лямбда):

хоча Java 8 включає підтримку функцій лямбда, вони не є першого класу; це означає, що, хоча можна використовувати лямбди для реалізації анонімних інтерфейсів з одним методом, Java не підтримує передачу функцій як аргументів методу. Крім того, відсутність умовиводу типу робить твердження лямбди досить незручним, особливо при спробі імітувати такі функції, як склад функцій або каррі; відсутність підтримки їх у   
мові [5];

* Прив'язка findViewById:

хоча ця проблема характерна для Android як платформи, а не Java як мови, також варто вказати на кількість типового коду, необхідного для отримання посилання на вигляд Android. Хоча взагалі вдається усунути громіздкий findViewById (int id) завдяки dataBinding, розробнику все одно доведеться зберігати посилання на цю прив'язку.

Java є дуже багатослівною мовою, вона вимагає написання великої кількості коду для будь-якої операції, а також створення великої кількості файлів (по одному на клас). Перша проблема може привести до коду, більш дорогого в обслуговуванні і схильного до помилок. Другий - проблема розповсюдження класів [6].

З плином часу потреба мати мову з реальною та рідною підтримкою для всього, що згадувалося вище, стає все необхіднішою, а також підтримка основної функції Android, її здатності, написання та компіляції одного додатка, змусити його працювати на будь-якому пристрої та його версії.

Література

1. Вонг, В. Напишіть один раз, налагоджуйте всюди. electronicdesign.com. – 2002. -[Електронний ресурс] – стаття. – Режим доступу: <https://web.archive.org/web/20090321180726/http://electronicdesign.com/Articles/Index.cfm?ArticleID=2255&pg=3>.
2. Tech Crunch Kotlin стає обраною мовою Google для розробки додатків для Android. - 2019. - [Електронний ресурс] – стаття. – Режим доступу: <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/>.
3. Що не так з Java: знищення типізації – 2006. - [Електронний ресурс] – стаття. – Режим доступу: [www.safalra.com/programming/java/wrong-type-erasure/](http://www.safalra.com/programming/java/wrong-type-erasure/).
4. Kotlin для кросплатформної мобільної розробки. JetBrains: інструменти розробника для професіоналів і команд – 2020. - [Електронний ресурс] – стаття. – Режим доступу: <https://kotlinlang.org/lp/mobile/>.
5. Філліпс Б. Android. Програмування для професіоналів / Філліпс Б. – М., 2017. – С. 308.
6. Глушаков С. В. Програмування на Java 2 - Харків: АСТ : Фоліо, 2001 – С. 137.