

# ПОРІВНЯННЯ ОПЕРАЦІЙНИХ СИСТЕМ IOS ТА ANDROID. ПЕРЕВАГИ ТА НЕДОЛІКИ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ КОЖНОЇ З СИСТЕМ

Якимчук В.С., к.т.н., доц. каф. БМК  
[viktoria.iakymchuk@gmail.com](mailto:viktoria.iakymchuk@gmail.com)

Гаврилюк В.С., студент  
[havryliuk.vitalik@gmail.com](mailto:havryliuk.vitalik@gmail.com)

Факультет біомедичної інженерії,  
Національний технічний університет  
«Київський політехнічний інститут імені Ігоря Сікорського»,  
м. Київ, Україна

*Реферат – Можливості мобільних додатків давно вийшли за рамки індустрії розваг. Їх все частіше використовують у сферах великого та малого бізнесу. Також потрібно підкреслити, що кожна компанія прагне мати власні мобільні додатки. Адже саме це привертає увагу нових клієнтів та робить надання послуг більш комфортними та ефективними для користувачів. Саме тому ринок потребує більшої кількості спеціалістів у сфері розробок мобільних додатків. Відповідно в кожного розробника мобільного додатку виникає питання: з якою операційною системою працювати. У них є можливість вибору серед двох основних напрямків розробки: iOS та Android. У кожній операційній системі є свої переваги та недоліки.*

*Ключові слова – iOS, Android, операційна система, мобільний додаток, Xcode, Android Studio, Swift, Objective-C, Kotlin, Java, IDE (integrated development environment), App Store.*

## I. ВСТУП

Останнім часом у світі стає все більше пристроїв на базі мобільних платформ, та все частіше з'являються нові. У цей час основна частина ринку розділена між двома мобільними платформами, що найбільш активно розвиваються: iOS (власницька мобільна операційна система від компанії Apple) та Android (операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux).

За своєю продуктивністю мобільні пристрої вже досягли рівня настільних комп'ютерів. Раніше до платформ, під керуванням яких працювали, наприклад, мобільні телефони, пред'являлися чітко встановлені вимоги. Невід'ємною частиною мобільних платформ був набір додатків, ретельно налагоджених, протестованих виробником і завантажених на мобільний пристрій разом із самою платформою. Завантаження сторонніх додатків було не широко поширене. Для нових типів мобільних платформ, навпаки, завантаження сторонніх додатків є однією з основних можливостей. Сторонні розробники

одержали можливість створення і встановлення своїх додатків на мобільні платформи.

Компанією Apple створено новий механізм масового поширення мобільних додатків – App Store. Механізм виявився успішним і був узятий на озброєння іншими компаніями (розробниками мобільних платформ). Цей механізм дозволив стороннім розробникам отримувати прибуток від створення й поширення мобільних додатків. Не залишилися у стороні й Android-розробники, які через декілька місяців презентували світу Android Market (зараз Google Play).

## II. ОГЛЯД ЛІТЕРАТУРИ

Розробка мобільних додатків у багатьох аспектах подібна до розробки додатків для настільних комп'ютерів. У той же час розробка мобільних додатків ставить перед програмістами нові завдання. До таких завдань, зокрема, відносять: більш уважне керування пам'яттю мобільної платформи, врахування обмеженого заряду батареї, а також можливої нестабільності бездротового мережевого з'єднання.

Одержання знань по розробці додатків для мобільних платформ може бути цікаво як з погляду застосування у професійній діяльності, так і з погляду більш глибокого розуміння роботи мобільного програмно-апаратного комплексу.

Мобільні пристрої на базі нових мобільних платформ, у тому числі смартфони, стали частиною повсякденного життя. Вони використовуються не лише для спілкування. Поряд зі стандартними можливостями мобільних телефонів, як-от: дзвінки, обмін текстовими повідомленнями через SMS або в соціальній мережі, – створюються додатки, що інтегрують функції обміну інформацією, які працюють одночасно з декількома соціальними мережами. Усе активніше відбувається обмін відеоінформацією за рахунок вбудованих у смартфони камер. У подорожах виявляється дуже затребувана функціональність Gps-навігатора (Global Positioning System, Глобальна система позиціонування). У розробників з'являється можливість використовувати власні додатки у своєму повсякденному житті, що дозволяє їм по-новому глянути на свій продукт. Під час навчання розробників мобільних додатків цей фактор може додатково мотивувати й збільшити інтерес до процесу створення.

У кожного мобільного розробника постає питання вибору платформи для розробки мобільних додатків. Є декілька основних критеріїв за якими можна зрозуміти, яку операційну систему обрати.

Неважно помітити, що сьогодні достатньо багато операційних систем, що розроблені для смартфонів та планшетних комп'ютерів. Кожна система має свої переваги та недоліки, індивідуальні характеристики та перелік пристроїв, на яких вона може бути встановлена.

Однією з перших тенденцій, що супроводжують швидкий розвиток мобільних платформ, став процес переносу на них практично всіх технологій, що були розвинуті для настільних комп'ютерів [1]. У розробці додатків намітилося кілька напрямків. Серед них варто відзначити ігрові додатки, додатки для роботи із соціальними мережами й бізнес додатки, орієнтовані на документообіг у середніх і великих компаніях. По способу розробки виділилося два основних види

додатків: “native” додатки й web-додатки. “Native” додатки створюються повністю на базі власного API мобільної платформи. Web-додатки включають клієнтський додаток, на основі невеликої кількості “native” коду, що одержує контент від серверного додатку, й відображує його з використанням стандартних web-технологій, у тому числі HTML (Hypertext Markup Language, Мова розмітки гіпертекстових документів), JavaScript (динамічна, об'єктно-орієнтована прототипна мова програмування). Серверний додаток може бути створений заново або може бути задіяний наявний серверний додаток з відомим API (Application Programming Interface, набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання у зовнішніх програмних продуктах).

Ігрові додатки пишуться на “native” коді для досягнення максимальної продуктивності. Для додатків, що працюють із соціальними мережами, створюється клієнтський додаток, що використовує для одержання даних стандартний API сервера тієї або іншої соціальної мережі. Для розробки бізнес додатків використовується як “native” код, так і web-технології.

Операційна система iOS випускається тільки для пристроїв, які виготовленні фірмою Apple: смартфон iPhone, медіа програвачів iPodTouch, смарт-годинника AppleWatch тощо [2]. iOS доволі проста у використанні, навіть найбільш непрофесійному користувачу буде легко в ній розібратися. Однак за зовнішньою простотою ховається доволі складна система [3].

Під час розробки мобільного додатка програміст повинен розуміти, як його додаток використовує пам'ять платформи. Кілька корисних підходів для зменшення використання пам'яті під час розробки мобільних додатків представлені в статті “On Mobile Java Memory Consumption” [4]. Зазначені підходи, в першу чергу, орієнтовані на платформу Java ME. Однак вони можуть бути поширені й на інші мобільні платформи.

Іншою проблемою для мобільних пристроїв є велика різноманітність їх екранів і розширень. Практичне рішення для створення “гнучкого” графічного інтерфейсу користувача

для платформи Java ME зазначено в статті “Better Midlets by design” [5].

Під час першої презентації iPhone, компанія Apple декілька разів наголосила, що мобільна платформа iOS заснована на UNIX-ядрі, заснованому в мікро ядрі Mach. Коли iOS вперше вийшла на ринок споживачів, вона не мала багатозадачності, можливості встановлювати сторонні додатки, не мала підтримки копіювання та вставки. Однак з часом iOS удосконалювалась і стала системою, яку можна порівнювати з деякими комп'ютерними операційними системами. iOS може працювати тільки на смартфонах та планшетних комп'ютерах з процесорами архітектури ARM. Інтерфейс користувача мобільної платформи iOS заснований на концепції прямої маніпуляції за допомогою жестів multitouch. iOS використовує той же набір основних елементів Darwin, який сумісний зі стандартом POSIX. В iOS є чотири шари абстрагування: Core OS, Core Services, Media Layer, Cocoa Touch [4-6].

В операційній системі iOS використовується мова Objective-C. Розробка на настільній операційній системі Mac OS X, у якості інтегрованого середовища розробки використовується Xcode. З Xcode інтегрований додаток для пошуку витоків і спостереження за виділенням пам'яті Instruments і додаток Interface Builder для створення графічного інтерфейсу. Для симуляції роботи додатку на мобільному пристрої служить iPhone Simulator. З використанням iPhone Simulator не можна перевірити роботу деяких специфічних API, наприклад, роботу з акселерометром, геопозиціонуванням і камерою. iPhone Simulator входить до складу iPhone SDK (Software Development Kit). iPhone SDK містить фреймворки й бібліотеки для складання додатків під реальний пристрій iPhone і iPhone Simulator. Xcode і iPhone SDK можна завантажити on-line [7] з офіційного сайту розробників додатків для платформ компанії Apple. Для завантаження інструментарію потрібна реєстрація. Реєстрація припускає участь в одній із програм для розробників. Найбільш простою є програма iOS Developer Program [8]. Вона безкоштовна й дозволяє завантажити й використовувати інструментарій розробки, а

також завантажувати й тестувати власні додатки на iPhone Simulator. Можливість завантаження додатків на реальний пристрій відсутня. Для одержання безкоштовної можливості завантаження й тестування на реальному пристрої можна використовувати програму iOS University Program. Однак потрібне підтвердження того, що учасник програми дійсно буде використовувати реальний пристрій iPhone лише з метою університетського навчання [9].

Swift – це нова мова програмування для розробки iOS та OSX додатків, яка поєднує у собі все найкраще з мов програмування C та Objective-C. Однак мова не має обмежень, які накладаються в угоду сумісності з C. У Swift використовуються патерни безпечного програмування і додані сучасні функції, що перетворюють створення додатку в простий, більш гнучкий та цікавий процес. Swift розроблявся декілька років. Основою нової мови програмування стало існування компілятора, фреймворки та відладчик. Розробники спростили процес управління пам'яттю за допомогою механізму автоматичного підрахунку посилай – Automatic Reference Counting (ARC).

Фреймворки також зазнали серйозної модернізації. Objective-C почав підтримувати літерали та модулі – все це створило сприятливі умови для впровадження сучасних технологій. Саме ця підготовча робота стала фундаментом для нової мови програмування, яка застосовується для розробки програмних продуктів Apple [10].

Розробникам Objective-C Swift буде знайомим. Він поєднує у собі читабельність іменованих параметрів і могутність динамічної об'єктної моделі Objective-C. Додаток відкриває доступ до вже наявного фреймворку Cocoa та сумісний з кодом, який написаний на Objective-C. Swift підтримує так звані playground-и, які дозволяють програмістам експериментувати з кодом, дивитися на результат у реальному часі без необхідності компілювати і запускати додаток [10].

В операційній системі Android для розробки додатків використовується Java. Можна написати код додатку для Android з допомогою Google API Java, який буде скопійований у файли класів. Android не

застосовує віртуальну машину Java (JVM) для виконання файлів класів, замість цього в ньому використовується віртуальна машина Dalvik. Вона не працює з Java-байткодом та не є істинною JVM. Для виконання на віртуальних машинах Dalvik, файли класів компілюються у форматі DEX (Dalvik EXecutable). Після перетворення у формат DEX, файли класів разом з іншими ресурсами об'єднуються у пакети Android (APK) для розповсюдження та інсталяції на різних приладах [10].

Головне, що потрібно знати: в основній базовій бібліотеці класів віртуальної машини Dalvik лежить підмножина проекту Apache Harmony, як наслідок вона не підтримує всі API J2SE [10].

У червні 2009 року вийшов перший офіційний реліз Android NDK (Native Development Kit). Розробники одержали можливість розробляти власні бібліотеки за допомогою мови C для платформи Android. Використання коду мовою C у критичних за продуктивністю ділянках додатка суттєво підвищує його швидкодію. Розробка може вестися на інструментальних машинах під керуванням Windows, Linux та Mac OS X. Існує два способи розробки мобільних додатків: з використанням консольних утиліт і з використанням ADT plug-in для Eclipse. Для імітації роботи на реальному пристрої Android служить емулятор Android, створений на базі віртуальної машини Qemu. Емулятор Android входить до складу Android SDK. Також в Android SDK входять засоби профілювання, налагодження й створення графічного інтерфейсу додатків. Завантажити Android SDK і Android NDK можна on-line [11] з офіційного сайту розробників для платформи Android. Також за цим лінком можна ознайомитись з покроковою інструкцією щодо установки інструментарію [9].

Використання інструментарію розробки під платформу Android емулятора, а також завантаження й тестування на реальному пристрої не вимагають оплати [9].

У липні 2011 року JetBrains розповіли про нову мову програмування, її назвали Kotlin. Ця мова програмування для JVM. Основна філософія мови полягає у тому, що Kotlin краща за Java. Kotlin схожа на Swift. Її переваги в лаконічності мови, розширення,

іменовані аргументи; мова підтримується Google, і вже вийшло декілька стабільних версій; мова повністю сумісна з Java; для того, щоб підключити Kotlin не потрібно все переписувати, можна робити нові частини на Kotlin. Мінусом цієї мови програмування є мала спільнота і повільне просування, попри те, що з'являються бібліотеки та JetBrains проводить різні заходи для залучення більшої аудиторії [12].

### III. МЕТА ДОСЛІДЖЕННЯ

Порівняти дві операційні системи (iOS та Android), вказати переваги та недоліки кожної з них. Надати порівняльну характеристику операційним системам, яка допоможе зробити вибір розробникам мобільних додатків, які розпочинають свій професійний шлях, під час вибору платформи для розробки мобільних додатків. Вирішення встановленої мети сприятиме у майбутньому правильному вибору щодо бажаної операційної системи та обранню того напряму роботи, який буде розвиватися надалі.

### IV. МАТЕРІАЛИ ДОСЛІДЖЕНЬ

Порівняння двох операційних систем (ОС) – iOS та Android – проводилося на основі опрацьованої літератури, створеного мобільного додатку для ОС iOS та опитування розробників, які створюють мобільні додатки під різні ОС. Опитування пройшли 10 мобільних розробників, віком від 20 до 27 років. Стаж роботи кожного працівника складає більше двох років. З 10 опитаних мобільних розробників були – 5 iOS та 5 Android розробників.

Розроблено мобільний додаток для ОС iOS для переводу документів в електронний формат (рис. 1).

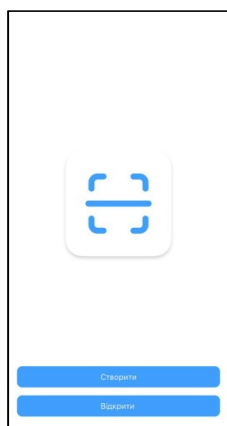


Рис.1. Початковий екран користувача

## V. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Оскільки практично всі сьогодні носять з собою смартфони та різні мобільні пристрої, мобільні програми дозволяють мати необхідну інформацію під рукою. Ці програми дозволяють ефективно інтегрувати інформацію з соціальними мережами, сайтом компаній, мультимедійним контентом і засобами комунікації.

Будь-яка необхідна інформація може бути представлена в одному додатку та актуалізована для конкретної людини, місця і часу. Такий спосіб концентрованого надання інформації на персональному мобільному пристрої дозволяє значно скоротити відстань між компанією, брендом і джерелом інформації з клієнтом, споживачем або співробітником. Це наділяє користувача додатка додатковими можливостями, які недоступні йому в умовах відсутності подібної програми. Компанія ж отримує можливість постійної комунікації з користувачем за допомогою одного каналу, одночасно включає у себе функції кількох. Наявність власних бренданих і корпоративних мобільних додатків, які мають різні функції, – це сьогодні просто необхідність для підвищення ефективності компанії в умовах сучасного ринку. Крім того, зараз розробником програми може стати абсолютно кожен.

Розробка мобільного додатку одразу для двох систем потребує більше часу та грошей. Якщо ресурси обмежені, то краще почати з однієї платформи. Щоб зробити правильний вибір потрібно взяти до уваги особливості iOS та Android.

Перше на що потрібно звернути увагу, це демографічні особливості. Android – найбільш популярна мобільна операційна система в світі. Ця платформа займає 86% на світовому ринку [13]. iOS друга за популярністю операційна система у світі. Незважаючи на те, що iOS не домінує на ринку, є регіони, де операційна система Apple лідирує. Це можна побачити в країнах з більш високим рівнем життя та розвиненим рівнем економіки, наприклад, у США та європейських країнах. Android популярний в розвинених регіонах Китаю [14].

Знання цільової аудиторії – один з найважливіших факторів у разі виникнення потреби розробити додаток. Інформація про вік, дохід, сімейний статус, професію та інтереси допоможе визначити, якій саме операційній системі віддають перевагу майбутні користувачі додатку.

Вартість розробки проекту багато в чому залежить від його складності й можливостей. Як правило, додатки для ОС iOS створювати набагато простіше та швидше тому що вони будуть працювати на невеликій кількості пристроїв на відміну від Android. Через велику кількість фрагментації Android, процес розробки та тестування ускладнюються.

В iOS обмежена файлова система, тому деякі ідеї неможливо реалізувати. Android пропонує розробникам більше свободи тому, що Google дає доступ до файлів операційної системи та обладнанню.

Виходячи з вищесказаного, можна впевнено сказати, що краще використовувати iOS. Причини обумовлені більшою платоспроможністю і лояльною аудиторією, а також швидкістю виходу на ринок. Розробка програми спочатку для Android має сенс, якщо цільова аудиторія в основному користується Android-смартфонами.

Далі перейдемо до більш детального порівняння двох вказаних ОС. Спершу розглянемо операційну систему Android. Kotlin чи Java? Kotlin – це новітня статична типізована мова програмування з відкритим вихідним кодом. Він може ефективно запускатися на віртуальній машині Java (JVM). Kotlin розроблений на основі JetBrains та офіційно підтримується Google. У даний час Kotlin використовується для створення Android-додатків такими лідерами бізнесу як

Pivotal, Atlassian, Pinterest, Evernote і Uber. Остання статистика Appbrain показує, що в сегменті топових додатків 2018 року Kotlin займає 25,3% ринку [15]. При цьому 40,76% нових інсталяцій додатків також припадають на додатки, написані на Kotlin.

Послідовний та інтуїтивний синтаксис Kotlin забезпечує підвищення продуктивності команд розробників. Для написання і розгортання мобільного забезпечення (деплоймента) програми потрібно менше часу і меншу кількість рядків у коді. Таким чином можна швидше отримати готовий додаток. Kotlin має 100% сумісність з Java. Іншими словами, методи Java можна викликати із Kotlin. Це є перевагою не тільки для розробників, але і для компаній, що мають велику кодову базу на Java.

Android-розробникам легко підтримувати код на Kotlin, оскільки ця мова підтримується у багатьох IDE (integrated development environment), включаючи Android studio, і в декількох інших SDK. Крім того, розробники можуть працювати з будь-яким звичним набором інструментів. Остання версія Kotlin має зворотну сумісність з усіма попередніми версіями. Це позбавляє Android-розробників від величезного «головного болю» – роботи з різними версіями. Kotlin вивчати легше, ніж Java, тому що для цього не потрібно ніяких знань у сфері розробки мобільних додатків. Android Studio надає розширену підтримку Kotlin та інструменти адаптації. Розробники можуть працювати одночасно на Kotlin і на Java.

До недоліків Kotlin відносять швидкість компіляції, менша підтримка спільноти, брак розробників (дуже обмежена кількість людей займаються розробкою на Kotlin, недолік талантів у розробці Android-додатків стосується Kotlin більшою мірою, ніж Java).

Java вважають кращою за Kotlin. За допомогою віртуальної машини Java програми на цій мові можуть запускатися практично в будь-якій системі. В результаті Java отримує лідерство серед Android-додатків. Завдяки великим суспільним групам на GitHub і StackOverflow розробники можуть отримати допомогу практично з будь-якої проблеми.

Застосування цієї мови не обмежується лише розробкою Android-додатків. Java прекрасно підходить і для розробки крос

платформних додатків. Оскільки Android розроблений на Java, ця мова має вже готові бібліотеки і SDK для полегшення процесу розробки.

Потрібно відмітити, що Java значно відстає від Kotlin у швидкості, Java вимагає більше пам'яті і, в порівнянні з іншими мовами, працює набагато повільніше. Також код на Java довгий, а отже, він вимагає більше часу для написання, у ньому може бути більше помилок і багів.

Перейдемо до розгляду iOS. Процес розробки мобільних додатків на Swift простіше у порівнянні з Objective-C. Код, що написано на Swift зручніше читати та він простіший для розуміння. Сама мова потребує менше коду, і це надає їй перевагу.

Наприклад, система повторів типів Swift понижує складність проміжного коду, в той час як розробники iOS додатків, що пишуть на Objective-C, повинні запам'ятовувати спеціальні символи (%s, %d, %@) та надавати списки змінних для заміни кожного з них. Також Swift включає модулі, які видаляють клас префіксів та має більш зрозумілий закритий синтаксис.

Один із головних переваг Swift полягає у можливості створювати гнучкі класи, що полегшує розробникам задачу: у випадку необхідності введення описання, потрібно використовувати протокол Printable. Аналогічно проходить процедура і з порівнянням.

Необхідно підкреслити, що Swift сучасна мова, яка адаптується під сучасні підходи інших мов. Наприклад, функція додавання двох рядків можна реалізувати за допомогою знаку «+», а Objective-C не має такої можливості. Описуючи функції Swift, потрібно відмітити опцію Playground. Завдяки їй розробники iOS додатків легко можуть побачити, як працює додаток. Також, у разі їх використання, не потрібно запускати емулятор, що дозволяє економити час розробників.

Таким чином, у даному контексті перемагає Swift. Тим не менш, якщо розглядати процес розробки iOS додатків загалом, то Objective-C пропонує велику кількість бібліотек та багато інших корисних речей. Також дві ці мови можуть йти поруч у додатку.

Не дивлячись на те, що Swift має певні недоліки (недостатня підтримка сторонніх IDE та наявність багів в Xcode, наприклад, XCode 6.3 не розпізнає юніт-тест Swift), він пропонує більше переваг. Swift є сучасною мовою програмування, включає багато функцій та можливостей, у певній мірі запозичує корисні речі з інших мов та ігнорує непотрібні.

На основі порівняння двох ОС для розробки мобільного додатку обрано ОС iOS. Основною ціллю додатку є переведення документів в електронний формат. Мобільний додаток реалізовано та вводиться в експлуатацію (рис. 2).

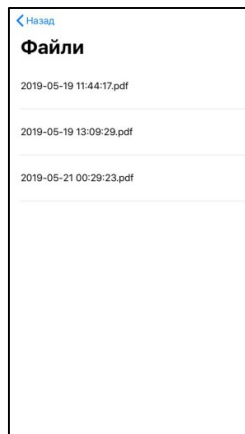


Рис. 2. Перелік створених електронних файлів

У програмі зберігається історія створення електронних документів, що дає змогу користувачу додатку передивитись всі збережені файли, швидко їх знайти та редагувати у бідь-який момент (рис. 3).

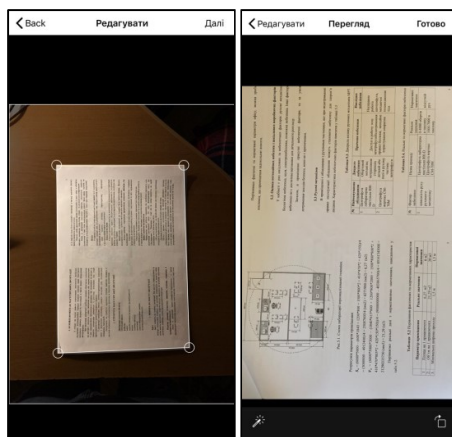


Рис.3. Екран редагування та екран перегляду зображення

Фотографуючи потрібну інформацію, користувач може використовувати один з режимів: автоматичний (програма розпізнає аркуш та обрізає фотографію одразу) та ручний (користувач фотографує та обрізає зображення власноруч). Слід зазначити, що під час використання будь-якого з двох режимів, користувач може додатково редагувати отримане зображення (обрізати зображення, додавати фільтри тощо).

Новий сфотографований електронний документ одразу відображається у списку створених раніше файлів. Щойно створений PDF-документ можна переглянути та, при необхідності, редагувати цей документ (рис. 4).

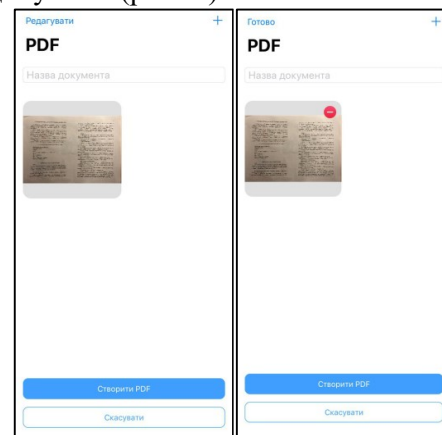


Рис. 4. Екран створення PDF-документа (звичайний режим та режим редагування)

У випадку, якщо користувач забажає видалити створений документ, буде з'являтися повідомлення з підтвердженням даної дії (рис. 5).

## VI. ВИСНОВКИ

Не можна бути впевненим на всі 100% у досконалості операційної системи або ж у мові програмування. У наш час все швидко розвивається і відмінністю кожної мови програмування та операційних систем – є їх швидкий розвиток та вдосконалення.

Порівняння двох операційних систем (ОС), iOS та Android, проводилося на основі опрацьованої літератури та опитування розробників, які створюють мобільні додатки під різні ОС.



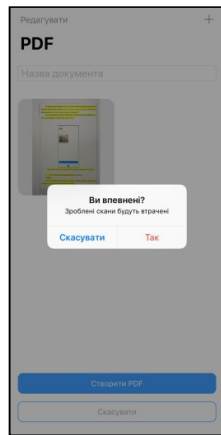


Рис.5. Вікно підтвердження або скасування видалення

Зазначимо, що певною мірою Swift представляє собою удосконалену версію Objective-C. Вона запозичила деякі сучасні функції та підходи з Objective-C та інших мов. Swift більш проста для розуміння та читання, потребує менше коду, витрачається менше часу для фіксування багів (помилки). Swift забезпечує безпечне управління пам'яттю, опцію Playground тощо. Swift швидша у порівнянні з Objective-C. І найголовніше, що мова постійно вдосконалюється.

На основі порівняння двох ОС для розробки мобільного додатку для переведення документа в електронний формат було обрано ОС iOS. Створений мобільний додаток є зручним та виконує всі потрібні функції для реалізації поставлених завдань. Наприклад, створений електронний документ можна надіслати (рис.6) за допомогою будь-якого методу (пошти, месенджера тощо).

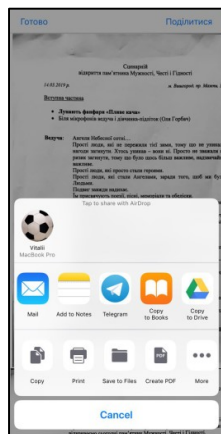


Рис. 6. Поширення PDF документу

## ПЕРЕЛІК ПОСИЛАНЬ

- [1] Mahmoud, Q.H., and Popowicz, P., “A Mobile Application Development Approach to Teaching Introductory Programming”, 40th ASEE/IEEE Frontiers in Education Conference, Session T4F, October 27-30, Washington, DC.
- [2] Hartikainen, V., Liimatainen, P.P., and Mikkokek, T., “On Mobile Java Memory Consumption.” Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-based Processing (PDP2006), Montbeliard-Sochaux, France, pp. 333-339.
- [3] Kontio, M., “Better MID lets by design.” IBM developer Works — [Электронный ресурс] — Режим доступа. — URL: [www.ibm.com/developerworks/wireless/library/wi-design.html](http://www.ibm.com/developerworks/wireless/library/wi-design.html)
- [4] Буга В.Д. ОПЕРАЦИОННАЯ СИСТЕМА IOS: ИСТОРИЯ, ПРОГРЕСС, СОВЕРШЕНСТВОВАНИЕ // XVIII Студенческая международная заочная научно-практическая конференция «Молодежный научный форум: технические и математические науки».
- [5] История iOS. Первое рождение легенды — [Электронный ресурс] — Режим доступа. — URL: <https://appleinsider.ru/istoriya-apple/1-istoriya-ios-pervoe-rozhdienie-legendy.html?preview=true>
- [6] От iPhone OS до iOS 7: история самой совершенной в мире мобильной платформы — [Электронный ресурс] — Режим доступа. — URL: <http://www.macdigger.ru/iphone-ipod/ot-iphone-os-do-ios-7-istoriya-samoj-sovershennoj-v-mire-mobilnoj-platfomy.html>.
- [7] iOSDevCenter — [Электронный ресурс] — Режим доступа. — URL: <https://developer.apple.com/devcenter/ios/index.action#downloads>
- [8] РОЗРОБКА ЗАСТОСУВАНЬ ДЛЯ МОБІЛЬНИХ ПРИБОРІВ // Конспект лекцій для студентів спеціальності 122 Комп'ютерні науки та інформаційні технології усіх форм навчання. 2016. – с. 55.
- [9] Как работает Android. Введение для Java-разработчиков — [Электронный ресурс] — Режим доступа. — URL: <https://javarush.ru/groups/posts/481-kak-rabotaet-android-vvedenie-dlja-java-razrabotchikov>
- [10] Введение в Swift: — [Электронный ресурс] — Режим доступа. — URL: <https://metanit.com/swift/tutorial/1.1.php>
- [11] Download the Android SDK: — [Электронный ресурс] — Режим доступа. — URL: <http://developer.android.com/sdk/index.html>
- [12] Почему следует выбрать Kotlin для следующего проекта на Android: — [Электронный ресурс] — Режим доступа. — URL: <https://oxozle.com/2017/08/23/pochemu-sleduet-vybrat-kotlin-dlya-sleduyushhego-proekta-na-android/>
- [13] Демографічні особливості — [Электронный ресурс] — Режим доступа. — URL: <https://www.macdigger.ru/news/post/ios-ili-android-dlya-kakoj-sistemy-razrabatyvat-prilozheniya>
- [14] «Android vs iOS»: Составлен рейтинг популярности двух платформ в разных странах — [Электронный ресурс] — Режим доступа. — URL: <https://vistanews.ru/computers/software/238457>
- [15] Kotlin — [Электронный ресурс] — Режим доступа. — URL: <https://www.appbrain.com/stats/libraries/details/kotlin/kotlin>



УДК 004.45

# СРАВНЕНИЕ ОПЕРАЦИОННЫХ СИСТЕМ IOS TA ANDROID. ДОСТОИНСТВА И НЕДОСТАТКИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ КАЖДОЙ ИЗ СИСТЕМ

Якимчук В.С., к.т.н., доц. каф. БМК

[viktoria.iakymchuk@gmail.com](mailto:viktoria.iakymchuk@gmail.com)

Гаврылюк В.С., студент

[havryliuk.vitalik@gmail.com](mailto:havryliuk.vitalik@gmail.com)

Факультет биомедицинской инженерии

Национальный технический университет

«Киевский политехнический институт имени Игоря Сикорского»,

г. Киев, Украина

*Реферат* – возможности мобильных приложений уже давно вышли за рамки индустрии развлечений. Их все чаще используют в сферах большого и малого бизнеса. Также нельзя не заметить, что каждая компания стремится иметь собственные мобильные приложения. Потому что это привлечет внимание новых клиентов, и появляется возможность намного удобнее и эффективнее предоставлять услуги для пользователей. Именно поэтому рынок требует большого количества специалистов в сфере разработки мобильных приложений. Следовательно, у каждого разработчика мобильного приложения возникает вопрос: с какой операционной системой работать? В своем распоряжении они имеют два основных направления работы: iOS и Android. У каждой операционной системы есть свои достоинства и недостатки.

*Ключевые слова* – iOS, Android, операционная система, мобильное приложение, Xcode, Android Studio, Swift, Objective-C, Kotlin, Java, IDE (integrated development environment), App Store.

UDC 004.45

# COMPARISON OF IOS AND ANDROID OPERATING SYSTEMS: ADVANTAGES AND DISADVANTAGES OF DEVELOPING MOBILE APPLICATIONS FOR EACH OF THE SYSTEMS

Iakymchuk V.S., PhD, Associate professor

[viktoria.iakymchuk@gmail.com](mailto:viktoria.iakymchuk@gmail.com)

Havryliuk Vitalii, student

[havryliuk.vitalik@gmail.com](mailto:havryliuk.vitalik@gmail.com)

Faculty of Biomedical Engineering

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Kyiv, Ukraine

*Abstract* – The capabilities of mobile applications have long gone beyond the entertainment industry. They are increasingly used in large and small business spheres. Also should emphasize that each company wants to have its own mobile applications. This is what attracts the attention of new customers and makes the provision of services more comfortable and effective for users. That is why the market needs more specialists in the field of mobile development. Therefore, every mobile developer has an issue to choose what operating system to work with. Therefore, there are two main spheres of development: iOS, Android. Each of them has its advantages and disadvantages.

*Key words* – iOS, Android, operating system, mobile application, Xcode, Android Studio, Swift, Objective-C, Kotlin, Java, IDE (integrated development environment), App Store.