

Тема 3. Програмні середовища для розробки мобільних додатків

Інтегроване середовище розробки (IDE – Integrated development environment) – комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора вихідного коду, інструментів для автоматизації складання та відналагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

Зі значним зростанням популярності хмарних технологій, з'явилися хмарні IDE, які дозволяють не прив'язуватись до конкретного комп'ютера.

Враховуючи те, що переважна більшість мобільних пристроїв використовують мобільні операційні системи Android та iOS (або похідні від них), варто зосередитись на програмних середовищах розробки саме для них.

3.1 Мови програмування

Перед тим, як визначитися з середовищем розробки для мобільних пристроїв необхідно визначитися з операційною системою та мовою програмування.

3.1.1 Мови програмування для Android

Java

Java, безперечно, є лідером у цій галузі та сміливо може називатися основною мовою розробки під Android. Той факт, що майже всі офіційні курси та навчальна документація з програмування під Android засновані цією мовою, свідчить про її популярність.

Для Java можна знайти велику кількість вихідного коду на GitHub, і самі розробники відзначають, що ця мова дуже зручна для написання мобільних додатків.

Java широко поширилася в десятках галузей програмування, тому розробникам Android-початківцям непогано було б почати з вивчення Java в контексті екосистеми Android.

Kotlin

Kotlin дозволяє використовувати всі найкращі сторони Java у програмному продукті без його недоліків. Синтаксис та інші функції програмування на Kotlin сучасні, прості для розуміння та швидкі у простому середовищі розробки.

У 2019 році Google назвав Kotlin пріоритетною мовою для платформи Android, що зробило її однією з офіційних мов програмування для розробки під Android. Крім того, вона повністю сумісна із Java.

Python

Python не має вбудованої підтримки на Android, але є деякі інструменти, які дозволяють розробляти програми Python, а потім перетворювати їх на APK. Найпопулярнішим інструментом (фреймворком) для перетворення коду Python на APK є проєкт Kivy.

Kivy був розроблений для прискорення розробки програм. Однак за відсутності нативної підтримки Python скористатися перевагами середовища Android буде неможливо. Програми, що використовують Kivy, скомпільовані у великі APK-файли, що може знизити їхню продуктивність.

Lua

Lua використовується для розробки кросплатформових мобільних програм, що працюють на Android. У зв'язку з Lua в Corona є спеціальне середовище розробки, яке пропонує користувачам набагато більш простий спосіб розробки програм для Android, ніж Android Studio, і забезпечує повний контроль над процесом.

Basic4Android (B4A)

Ця мова – найкращий вибір для новачка. Ви зможете використовувати його у середовищі розробки B4A. Сам синтаксис Basic4Android дуже нагадує популярний Visual Basic. При розробці програм на ньому ви зможете використовувати безліч додаткових бібліотек з різним функціоналом, і вам не потрібні спеціальні інструменти runtime для запуску програм.

Basic4Android має потужний візуальний конструктор, який підтримує різні орієнтації та розміри екрану. Ви можете навчитися багатьом новим речам без

особливих зусиль і створювати базові програми, але для створення просунутих програм вам потрібно вивчити одну з вищезгаданих мов.

3.1.2 Мови програмування для iOS

Objective-C

Objective-C – це мова програмування iOS, яку було розроблено ще у 80-х роках минулого століття. Тривалий час вона була найпопулярнішою мовою розробки компанії Apple. З'явилася вона шляхом схрещування C та Smalltalk. Тому її спочатку вважали лише надбудовою над мовою C, яка дозволяє розширити можливості.

Код підтримується, регулярно оновлюється, при цьому внесення змін просте. Багато документації та технічної літератури, велика спільнота, де люди допомагають один одному опанувати цю мову. Вона дуже схожа на будь-яку іншу з сімейства C, потрібно лише вивчити синтаксис. Крім того, Objective-C можна застосовувати всередині проєктів, написаних на Swift, оскільки вони сумісні.

Якщо користувач лише починає опанувати програмування під iOS, виникнуть складності з розумінням, вона складна у вивченні. Продуктивність не надто швидка, а сам процес складання уповільнюється, якщо відбувається взаємодія з файлами Swift, тому що використовується своєрідний «адаптер», що переводить код зі Swift до Objective-C.

Але навіть з усіма недоліками Objective-C є дуже популярною мовою програмування iOS. Вона використовується для великих проєктів та інтернет-магазинів.

Swift

Swift – це сучасна мова програмування для iOS, яка використовується більшістю розробників. Вона з'явилася у 2014 році, і увібрала в себе найкращі якості мов C та Objective-C. Головна особливість цієї мови полягає в тому, що на етапі програмування значно зменшується кількість помилок внаслідок суворої

типізації об'єктів. Додано велику кількість сучасних функцій: замикання, дженерики та інші.

Вона дуже швидка, навігація по файлах зрозуміла та спрощена, повністю побудована на C, тому легко читається. Зовні вона максимально нагадує англійську мову, її синтаксис спрощений та зрозумілий. Розмір коду зменшується завдяки лаконічності. Існує можливість використовувати шаблони, вона легко поєднується з Objective-C та має відмінний рівень безпеки. Використання динамічних бібліотек дозволяє додаткам працювати швидше та стабільніше.

Через те, що Swift – це молода мова програмування iOS, постійно відбуваються оновлення та зміни, тому доводиться ретельно стежити за готовими додатками. Залишається ще багато розробників, які використовують Objective-C, оскільки вона стабільніша.

Swift підходить для проектів різної складності, від інтернет-магазинів до банківських сервісів.

3.1.3 Універсальні мови

C#

Мова C# взяла найкраще з Java та додала кілька своїх цікавих функцій. Програмування під на C# має на увазі використання Xamarin. Його можна використовувати для створення однієї логіки програми на C# для Android та iOS.

Що стосується розробки під Android, то можна виділити наявність доступу до найбільш функціональних середовищ розробки: Visual та Xamarin Studio. C# також корисна, коли вам потрібно використовувати Unity – кроссплатформний ігровий рушій для створення 2D або 3D-ігор для мобільних пристроїв.

C++

C++ – це кроссплатформна мова програмування, яку можна використовувати для створення високопродуктивних мобільних та настільних додатків. Ця універсальна мова високого рівня була представлена як розширена версія мови програмування.

Це дає розробникам повний контроль над пам'яттю та системними ресурсами. Для розробки Android- та iOS-програм можна використовувати велику кількість бібліотек, доступних для цієї мови.

Dart

Dart пропонує єдиний підхід до розробки для iOS та Android з використанням Flutter. Flutter дозволяє розробляти кросплатформні програми; тому вона ефективна при розробці програм для Android із загальною кодовою базою мовою Dart.

Dart позиціонується як альтернатива JavaScript. Компілюється в бінарний код, за рахунок чого досягається висока швидкість виконання операцій. Це технологія від Google, повна SDK, випущена в 2018 році. По ряду параметрів вважається найкращим рішенням, ніж React Native.

JavaScript

JavaScript – дуже популярна мова, яка дозволяє веб-розробникам робити все, від поліпшення інтерфейсу своїх вебсайтів до створення повних веб-додатків.

Деякі фреймворки JavaScript спеціально призначені для мобільних платформ розробки, таких як Ionic та React Native. За допомогою цих фреймворків та бібліотек легко розробляти кросплатформні мобільні програми. Це означає, що вам потрібно написати лише одну версію програми, і вона буде працювати на iOS або Android.

3.2 Середовища розробки

3.2.1 Android Studio

Android Studio – інтегроване середовище розробки (IDE) для платформи Android, представлено 16 травня 2013 року на конференції Google I/O менеджером по продукції корпорації Google – Еллі Паверс (англ. Ellie Powers). 8 грудня 2014 року компанія Google випустила перший стабільний реліз Android Studio 1.0.

Android Studio прийшло на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі вихідного коду продукту IntelliJ IDEA Community Edition, що розвивається компанією JetBrains. Android Studio розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0.

Бінарні складання підготовлені для Linux (для тестування використаний Ubuntu), macOS і Windows. Середовище надає засоби для розробки застосунків не тільки для смартфонів і планшетів, але і для носимих пристроїв на базі Wear OS, телевізорів (Android TV), окулярів Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для застосунків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

Для прискорення розробки застосунків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компоновання, що надає зручний попередній перегляд різних станів інтерфейсу застосунку (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовище вбудовані функції завантаження типових прикладів коду з GitHub.

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.

Деякі особливості будуть пізніше розгорнуті для користувачів так як програмне забезпечення розвивається; наразі, передбачені такі функції:

- Живі макети (layout): редагувальник WYSIWYG – живе кодування – подання (rendering) програми в реальному часі.
- Консоль розробника: підказки по оптимізації, допомога по перекладу, стеження за напрямком, агітації та акції — метрики Google аналітики.
- Резерви бета релізів та покрокові релізи.
- Базування на Gradle.
- Android-орієнтований рефакторинг та швидкі виправлення.
- Lint утиліти для охоплення продуктивності, юзабіліті, сумісності версій та інших проблем.
- Використання можливостей ProGuard та підписів до програм.
- Шаблони для створення поширених Android дизайнів та компонентів.
- Багатий редактор макетів (layouts) що дозволяє користувачам перетягнути і покласти (drag-and-drop) компоненти користувацького інтерфейсу, як варіант, переглянути одночасно макети (layouts) на різних конфігураціях екранів.

3.2.2 Eclipse

Eclipse – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation і включає проекти, такі як платформа Eclipse, набір інструментів для програмістів на мові

Java, системи контролю версій, конструктори GUI тощо. Написаний в основному на Java, може бути використаний для розробки застосунків на Java і, за допомогою різних плагінів, на інших мовах програмування, включаючи Ada, C, C++, C#, COBOL, Fortran, Perl, PHP, Python, R, Ruby (включно з каркасом Ruby on Rails), Scala, Clojure та Scheme. Середовища розробки зокрема включають Eclipse ADT (Ada Development Toolkit) для Ada, Eclipse CDT для C/C++, Eclipse JDT для Java, Eclipse PDT для PHP.

Початок коду йде від IBM VisualAge, він був розрахований на розробників Java, складаючи Java Development Tools (JDT). Але користувачі могли розширяти можливості, встановлюючи написані для програмного каркаса Eclipse плагіни, такі як інструменти розробки під інші мови програмування, і могли писати і вносити свої власні плагіни і модулі.

Випущена на умовах Eclipse Public License, Eclipse є вільним програмним забезпеченням. Він став одним з перших IDE під GNU Classpath і без проблем працює під IcedTea.

Eclipse це фреймворк для розробки модульних платформонезалежних застосунків із низкою особливостей:

- можливість розробки ПЗ на багатьох мовах програмування (рідною є Java);
- платформонезалежна;
- модульна, призначена для подальшого розширення незалежним розробниками;
- з відкритим сирцевим кодом;
- розробляється і підтримується фондом Eclipse, куди входять такі постачальники ПЗ, як IBM, Oracle, Borland.

Спочатку проєкт розроблявся в IBM як корпоративний стандарт IDE для розробки на багатьох мовах під платформи IBM. Потім проєкт було перейменовано на Eclipse і надано для подальшого розвитку спільноті.

Eclipse насамперед повноцінна Java IDE, націлена на групову розробку, має засоби роботи з системами контролю версій (підтримка CVS входить у

поставку Eclipse, активно розвиваються кілька варіантів SVN модулів, існує підтримка VSS та інших). З огляду на безкоштовність, у багатьох організаціях Eclipse – корпоративний стандарт для розробки ПЗ на Java.

Друге призначення Eclipse – служити платформою для нових розширень. Такими стали C/C++ Development Tools (CDT), розроблювані інженерами QNX разом із IBM, засоби для підтримки інших мов різних розробників. Безліч розширень доповнює Eclipse менеджерами для роботи з базами даних, серверами застосунків та інших.

З версії 3.0 Eclipse став не монолітною IDE, яка підтримує розширення, а набором розширень. У основі лежать фреймворки OSGi, і SWT/JFace, на основі яких розроблений наступний шар – платформа і засоби розробки повноцінних клієнтських застосунків RCP (Rich Client Platform). Платформа RCP є базою для розробки різних RCP програм як торент-клієнт Azareus чи File Arranger. Наступний шар — платформа Eclipse, що є набором розширень RCP — редактори, панелі, перспективи, модуль CVS і модуль Java Development Tools (JDT).

Eclipse написана на Java, тому є платформонезалежним продуктом, крім бібліотеки графічного інтерфейсу SWT, яка розробляється окремо для більшості поширених платформ. Бібліотека SWT використовує графічні засоби платформи (OS), що забезпечує швидкість і звичний зовнішній вигляд інтерфейсу користувача.

3.2.3 NetBeans

NetBeans IDE – вільне інтегроване середовище розробки (IDE) для мов програмування Java, JavaFX, C/C++, PHP, JavaScript, HTML5, Python, Groovy. Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації. Середовище розробки NetBeans за замовчуванням підтримує розробку для платформ J2SE і J2EE.

Поширюється у початкових текстах під ліцензією Apache License. Проєкт NetBeans IDE підтримувався і спонсорувався фірмою Sun Microsystems і після

придбання Sun – Oracle. У жовтні 2016 року Oracle передав NetBeans у власність Apache Software Foundation, яка займається розробкою і підтримкою проєкту.

NetBeans IDE доступна для платформ Microsoft Windows, GNU/Linux, FreeBSD, і Solaris (як SPARC, так x86). Для інших платформ доступна можливість зібрати NetBeans самостійно із початкових текстів.

За якістю і можливостям останні версії NetBeans IDE змагається з найкращими інтегрованими середовищами розробки для мови Java, підтримуючи рефакторинг, профілювання, виділення синтаксичних конструкцій кольором, автодоповнення мовних конструкцій на льоту, шаблони коду та інше.

У випуску 7.4 у жовтні 2013 продовжено розвиток засобів для розробки з використанням технологій HTML5, додана підтримка створення гібридних HTML5-застосунків для платформ Android і Apple iOS з використанням фреймворку Apache Cordova, реалізовані засоби використання HTML5 в проєктах Java EE і PHP, представлена експериментальна підтримка майбутнього випуску JDK8.

3.2.4 Microsoft Visual Studio

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також вебсайти, вебзастосунки, вебслужби як у рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Версія Visual Studio, під кодовим ім'ям Dev14 була представлена 20 червня 2015 року. Суттєвою зміною стала підтримка багатьох цільових платформ: окрім базової Windows з'явилась можливість будувати проєкти для IOS та Android.

3.2.5 Xcode

Xcode – інтегроване середовище розробки (IDE) виробництва Apple. Дозволяє створювати програмне забезпечення з використанням таких технологій як GCC, GDB, Java та ін. На сьогодні є єдиним засобом написання «універсальних»(Universal Binary) прикладних програм для Mac OS X.

Xcode включає в себе більшу частину документації розробника від Apple та Interface Builder – застосунок, який використовується для створення графічних інтерфейсів.

Пакет Xcode містить змінену версію вільного набору компіляторів GNU Compiler Collection і підтримує мови C, C++, Objective-C, Swift, Java, AppleScript, Python і Ruby з різними моделями програмування, включаючи (але не обмежуючись) Cocoa, Carbon і Java. Сторонніми розробниками реалізована підтримка GNU Pascal, Free Pascal, Ada, C#, Perl, Haskell і D. Пакет Xcode використовує GDB як back-end для свого налагоджувача.

У серпні 2006 Apple оголосила про те, що DTrace, фреймворк динамічного трасування від Sun Microsystems, випущений як частина OpenSolaris, буде інтегрований в Xcode під назвою Xray. Пізніше Xray був перейменований в Instruments.

3.2.6 Qt Creator

Qt Creator – інтегроване середовище розробки, призначене для створення крос-платформових застосунків з використанням бібліотеки Qt. Підтримується розробка як класичних програм мовою C++, так і використання мови QML, для визначення сценаріїв, в якій використовується JavaScript, а структура і параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt Creator може використовувати GCC або Microsoft VC++ як компілятор і GDB як зневаджувач. Для Windows версій бібліотека комплектується компілятором, заголовними і об'єктними файлами MinGW.

Сирцеві тексти Qt Creator поширюються в рамках ліцензії LGPL.

Qt Creator розроблена норвезькою компанією Trolltech, яку у 2008 році поглинула Nokia. Анонс проєкту відбувся на Qt Developer Days в жовтні 2008 року. Публічна бета-версія проєкту була опублікована 30 жовтня 2008. Фінальний реліз відбувся 3 березня 2009 року (разом з виходом Qt 4.5).

Після укладення стратегічного союзу з Microsoft Nokia втратила інтерес до розвитку технологій Qt. У березні 2011 фінська компанія Digia, постачальник ERP-систем, послуг і рішень в області мобільних систем і користувацьких інтерфейсів, оголосила про укладення угоди з Nokia про викуп у тої прав на комерційне ліцензування та надання послуг з підтримки розробки з використанням бібліотеки Qt. У вересні 2012 Nokia повністю відмовилася від Qt і Digia купує у Nokia весь бізнес і програмні технології, пов'язані з Qt.

3.2.7 Thunkable

Thunkable – візуальне середовище розробки програм для мобільних пристроїв. Слово «візуальне» означає, що програмування здійснюється візуальними блоками, що уособлюють функції та змінні. Бажано знати, що таке змінна та її область видимості, що таке функція та як працюють стандартні цикли, такі як for та while. Але в цілому конструктор зроблений настільки яскраво і по-дитячому, що з ним без проблем розбереться школяр, якого на інформатиці навчають бейсику. Ця візуальна мова називається «скретч» (scratch) і розроблена була спеціально для школярів.

Thunkable побудований на базі більш відомого MIT App Inventor 2, відкритій платформі, тому що заснованих на ньому безкоштовних конструкторів такого типу є багато, крім Thunkable. Але Thunkable пішли трохи далі за копіювання App Inventor, вони повністю переробили дизайн як самого конструктора, так і дизайн програми на виході з конструктора, зробили його більш сучасним. У Thunkable дуже кльове ком'юніті, правда тільки англійською. Також є дуже докладна документація щодо всіх інструментів платформи. Також у місцевого ком'юніті є своя бібліотека розширень, які додають у конструктор нові блоки та можливості. Всі платформи, що базуються на App Inventor, мають

один формат файлів проекту та розширень та у 95% сумісні один з одним. Це означає, що ви можете відкрити Thunkable будь-який проект, створений на App Inventor, AppyBuilder та інших платформах. Крім розширень (extensions) від ком'юніті тункабл ви можете використовувати розширення з бібліотеки App Inventor.

Існує дві версії Thunkable: класична та X (ікс).

Версія X більш нова, виглядає сучасніше і найголовніше – вона кросплатформова, тобто дозволяє створювати один додаток для iOS і Android. Однак вона не підтримує розширення та приклади з колишніх версій.