

УДК 004.023

Мельник В.М., Вознюк О.В.

Луцький національний технічний університет

ПРОГРАМА ІНТЕРВАЛЬНОГО ТАЙМЕРА ЗАСОБАМИ ANDROID STUDIO

Мельник В.М., Вознюк О.В. Програма інтервального таймера засобами Android Studio. Розроблено програмне забезпечення інтервального таймера для можливості контролю тренувань, яке буде працювати на операційній системі Android. Мобільний додаток може служити інтервальним таймером для великої кількості людей, які займаються спортом, та допоможе у всіх сферах де потрібен точний контроль за часом.

Ключові слова: android, sqlite, програмне забезпечення, додаток, база даних.

Мельник В.М., Вознюк О.В. Программа интервального таймера средствами Android Studio. Разработано программное обеспечение интервального таймера для возможности контроля тренировок, которое будет работать на операционной системе Android. Мобильное приложение может служить интервальным таймером для большого количества людей, которые занимаются спортом, та поможет во всех сферах, где нужен точный контроль за временем.

Ключевые слова: android, sqlite, программное обеспечение, приложение, база данных.

Melnyk V.M, Voznyuk O.V. Software interval timer means Android Studio. The software interval timer was developed for control trainings ability, that running on operating system Android. The mobile application can serve as the common interval timer for a large number of people involved in sports, and to help in all areas where need a strict control of the time.

Keywords: android, sqlite, software, application, database.

Вступ. Android – це операційна система для різноманітних пристроїв починаючи від смартфонів, планшетів, годинників – до окулярів “Google Glass”, телевізорів та інших. Після виходу в світ першої версії 23 вересня 2008 року дана операційна система активно розвивалась і отримала багато її оновлень, кожна версія якої отримує своє особисте кодове ім'я на тему десертів [1].

Станом на 2015 рік операційна система Android була встановлена на 82% мобільних пристроях [2]. Оскільки дана операційна система є найпоширенішою у сегменті мобільних пристроїв, тому доцільним є розробка мобільного програмного забезпечення, саме для даної операційної системи. Велика увага, заодно, приділяється також розробці програмного забезпечення для операційної системи, чому і присвячена дана робота.

Постановка проблеми. У ході роботи було поставлене завдання створення мобільного додатка, а саме – таймера для пристроїв, які працюють з підтримки операційної системи Android. Для розробки додатка використовується мова програмування Java та середовище розробки Android Studio.

Аналіз існуючих рішень. На даний момент розробка додатків для Android мовою програмування Java ведеться за допомогою декількох відомих інструментів розробки:

- Eclipse, у якому реалізована підтримка Android за допомогою плагіну.
- IntelliJ IDEA – середовище розробки на мові Java, у якому додано підтримку Android.
- Android Studio – це середовище програмування, яке розроблено на основі IntelliJ IDEA та надається компанією Google як середовище за замовчуванням.

Оскільки компанія Google оновлює свій продукт відповідно до появи нових функцій, то у даній роботі використано саме Android Studio як інструмент розробки таймера.

У ході роботи виконана розробка програмного забезпечення – інтервального таймера. Основним цільовим призначенням таких таймерів є контроль тренувань. Існує декілька поширених рішень для такого виду таймерів:

1. Таймер «Табата» – це найпростіший серед інтервальних таймерів [3]. Принцип його роботи полягає у тому, що два окремо взяті таймери чергуються один з одним, змінюючи фазу тренування на фазу відпочинку. У даному виді інтервального таймера можливо запрограмувати час кожної із двох фаз та кількість чергувань.
2. Інтервальний таймер – що являє собою більш загальне поняття. На відміну він «табата» таймера, у цьому виді інтервального таймера можливе програмування усіх трьох частин: кількості фаз, їх тривалість, та кількість чергувань [4].

У ході цієї роботи буде розроблено інтервальний таймер другого виду та вирішено деякі його недоліки, а саме неможливість комплексного програмування роботи інтервального таймеру.

Виділення невирішених раніше проблем. У квітні 2016 року компанією Google було випущено нову версію середовища розробки Android Studio версії 2.0, у якому було покращено роботу емулятора Android, на якому проводиться тестування додатку, що розробляється, та покращено автоматично

згенерований код для покращення роботи самого додатка [5]. Тому для розробки додатка було використано найновішою версією, яка дозволяє етапи розробки на більш якісному рівні.

Кожен із вказаних вище таймерів має свої недоліки. Як було вказано вище головним недоліком таймера «табата» є неможливість гнучкого програмування, тобто неможливо змінити кількість фаз роботи. Здавалось би, другий вид повинен цілком влаштовувати як засіб програмування, але він також має не зовсім очевидний недолік.

Доволі поширеними є бігові програми в яких тренування може поділитися на декілька частин. В першій частині тренування проходить по одній програмі, тобто вказано кількість фаз, їх тривалість та кількість підходів. В другій частині тренування ці всі параметри змінюються, незважаючи на те що дане тренування є комплексним, а тому перепрограмування таймеру другого виду не є можливим прямо під час тренування [6]. Для вирішення цієї проблеми було запропоновано ввести додаткове поняття, а саме «розклад». «Розклад» – це структурний об'єкт, який об'єднує в собі вказану кількість інтервальних таймерів другого виду, що виконують свою роботу один за одним. Це дозволяє дуже гнучко запрограмувати всю його роботу. Дане рішення повністю вирішує проблему комплексних тренувань, знімаючи з користувача необхідність перепрограмувати інтервальний таймер під час тренування.

У програмному коді додатку «розклад» реалізовується як двовимірний масив цілочисельних значень, у якому кожен підмасив являє собою інтервальний таймер другого виду і містить у собі значення кількості чергувань та часові відрізки для кожної фази.

Виклад основного матеріалу роботи. Однією з основних проблем розробки програмного забезпечення для операційної системи Android є різноманітність версій самої системи (рис. 1).

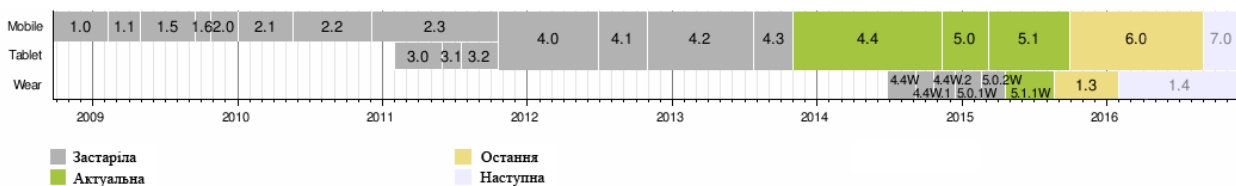


Рис. 1 – Оновлення версій Android

Згідно з даними, наданими Android Studio, вже 70,9% пристроїв працює на актуальній версії та вище, тобто починаючи з версії 4.4 під кодовим іменем «KitKat». Як видно з рисунка 1, досить великий сегмент пристроїв не буде підтримуватись якщо мінімальною версією для нашого додатку буде обрано версію 4.4, тому як мінімальну версію операційної системи для даного програмного забезпечення обрано версію 4.0 під кодовим іменем «IceCreamSandwich». Згідно з даними, які надає Android Studio, додаток буде підтримуватись на 97,3% пристроїв.

Іншою головною проблемою розробки програмного забезпечення для операційної системи Android є велика фрагментація екранів пристроїв. Згідно з даними, наданими компанією Google (рис. 2), оберемо оптимальний пристрій для розробки програмного забезпечення [7].

Отож, згідно з усіма даними, наведеними вище, розробка проводилась для пристроїв, починаючи з версії Android 4.0 «IceCreamSandwich» та для екранів normal hdpi, які встановлені на 41.5% пристроїв.

В даній роботі створено додаток інтервального таймера, який містить меню для створення таймерів та їх збереження до бази даних, а також меню вибору таймера з бази даних та екран безпосередньо з таймером, де відображається поточний прогрес.

Існує два основних підходи для компоновання елементів на екрані Android: розміщення, безпосередньо, компонентів на екран, та компоновання їх за допомогою фрагментів. В нашому додатку будемо користуватись підходом з використанням фрагментів.

Додати фрагмент на екран можливо двома способами. У першому випадку використовується розмітка, яка знаходиться в layout-файлі та безпосередньо розміщується фрагмент також у layout-файли. Та другий випадок, де використовується заміщення батьківського контейнера, створюється батьківський контейнер, наприклад Linear Layout, в який потім динамічно розміщується фрагмент. У додатку будемо використовувати другий спосіб, так як перший спосіб несе в собі певні недоліки, а саме неможливість динамічного оновлення вмісту фрагмента. Подібно до створення активності, для фрагментів створюється клас та xml-файл інтерфейсу, який потім підключається в методі onCreateView(), на відміну від схожого метода-події в активності onCreate(). Також, на відміну від активностей фрагменти не потрібно реєструвати в AndroidManifest-файлі.

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	2.4%						2.4%
Normal		4.9%	0.1%	41.5%	23.9%	14.9%	85.3%
Large	0.3%	4.6%	2.2%	0.5%	0.5%		8.1%
Xlarge		3.2%		0.3%	0.7%		4.2%
Total	2.7%	12.7%	2.3%	42.3%	25.1%	14.9%	

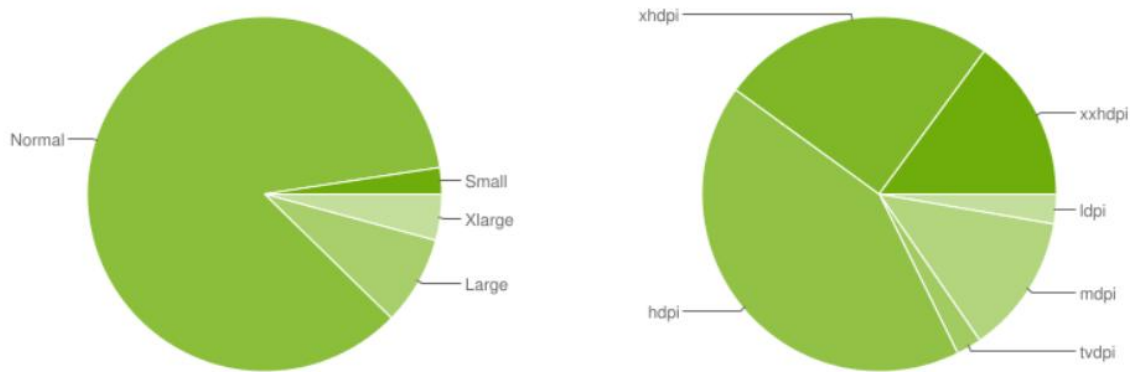


Рис. 2 – Фрагментація екранів пристроїв

В корені кожного додатку для Android повинен знаходитись файл маніфесту, який містить в собі важливу інформацію про додаток Android. Тільки після того, як буде отримана інформація з цього файлу, буде можливе виконання основного коду додатка [8].

Цей файл виконує і інші функції:

- Задає ім'я пакета Java-дodatка. Це ім'я служить унікальним ідентифікатором додатку.
- Описує компоненти додатку. Містить імена класів які реалізують кожен компонент, та публікує їх можливості. На основі цих декларацій система Android може визначити, з яких компонентів складається додаток і за яких умов їх можна запустити.
- Об'являє дозволи, які необхідні додатку.

Відрахунок часу в додатку здійснюється за допомогою класу `CountDownTimer`, який містить методи для роботи з ним, а саме `start()`, `cancel()`, `onTick()`, `onFinish()` [9]. Останні два методи потрібно реалізувати самостійно, що дозволяє запрограмувати логіку виконання таймера. Час для таймера зберігається в двовимірному масиві, де кожен масив представляє собою інтервальний таймер, у якому уже вказуються кількість повторів та час для виконання.

Окрім відображення часу таймера, прогрес відображається на екрані за допомогою кругового індикатора, що реалізований за допомогою класу `CircularProgressBar`, який є наслідником класу `View`. Тому було додано цей елемент безпосередньо у фрагмент, і у методі `onCreateView()` створюється об'єкт для цього `View`-елемента, а за допомогою вбудованих методів вказується колір індикатора.

Після завершення кожного відрізка часу лунає звуковий сигнал, а після закінчення роботи усього «розкладу» – лунає сигнал, відмінний від першого. Даний функціонал реалізується за допомогою класу `MediaPlayer`, тобто створюється об'єкт даного класу та виконується робота з ним, а саме відтворення wav-файлів, які знаходяться у директорії `raw`.

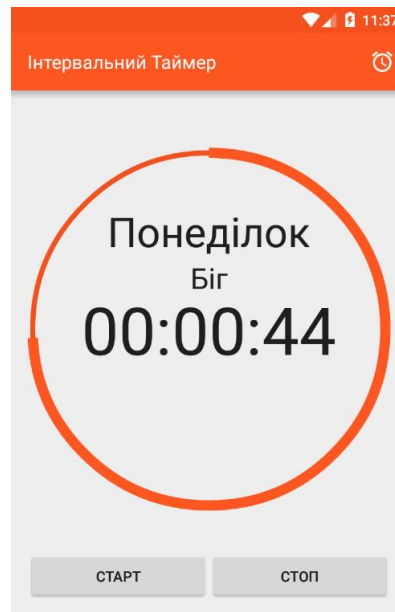


Рис. 3 – Інтерфейс головного екрану

База даних SQLite доступна відразу на будь-якому пристрої Android. В додатку, при підключенні до бази даних потрібно вказати ім'я бази даних та версію. При цьому може виникнути декілька ситуацій:

1. Ситуація, коли база даних не існує, наприклад при першому встановленні додатку. У цьому випадку потрібно створити базу даних та всі таблиці у ній і надалі працювати з даною базою даних.
2. Ситуація, коли база даних уже існує, але її версія застаріла, тобто потрібно оновити базу даних.
3. Ситуація, коли база даних створена з підтримки актуальної версії, тоді додаток просто підключається до неї.

У випадку, коли виникає ситуація 1 або 2 із вказаних вище, потрібно створити клас, який є наслідником для класу SQLiteOpenHelper та реалізувати методи: onCreate(), який буде викликатися, коли бази даних не існує, та метод onUpgrade(), який буде викликатися для оновлення бази даних при умові, коли її версія застаріла [10].

SQLite підтримує такі типи даних, як INTEGER – для цілочисельних значень, REAL – для значень з плаваючою крапкою, TEXT – для символічної стрічки у кодуванні UTF-8, UTF-16BE або UTF-16LE, NUMERIC – для збереження логічних значень, а також тут можна зберігати дату і час та BLOB, в якому можна зберігати бінарні дані [11].

Для нашого додатка потрібна не складна база даних в якій буде всього 3 поля. В них буде зберігатись ім'я «розкладу», імена інтервальних таймерів та сам «розклад». Для зберігання імен будемо використовувати тип TEXT, а сам «розклад», як було сказано вище, буде являти собою двовимірний масив, який не можна просто так помістити в базу SQLite. Тому вирішено було реалізувати метод-конвертер, що буде надавати можливість конвертації двовимірного масиву в стрічку, а також потрібну реалізацію метода, який буде здійснювати обернену конвертацію. Таким чином буде здійснюватися запис даних до бази даних.

Зчитування з бази даних відбувається на екрані вибору таймера. При створенні активності завантажуються список «розкладів» та будується список, з якого можна обрати «розклад» для роботи. Після цього дані передаються в головну активність та конвертуються у двовимірний цілочисельний масив.

Для надавання додатку привабливого естетичного вигляду використовується створена нами тема на базі вбудованої теми Theme.AppCompat.Light, для якої ми перевизначили вигляд ActionBar. Дані зміни вносимо в файл themes.xml. Структура цього файлу наступна:

- <resources> – кореневий елемент у який поміщаються стилі;
- <style> – елемент стилю, у якому вказується ім'я стилю та елементи <item>;
- <item> – конкретний елемент, для якого задається оформлення.

Потім у файлі styles.xml вибираємо активною нашу тему та основні кольори, що використовуються у додатку.

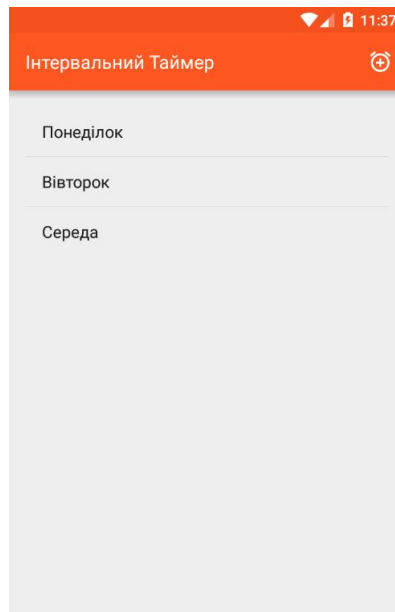


Рис. 4 – Екран вибору «розкладу» таймера з бази даних

У додатку використовується тема з використанням меню `ActionBar`, на якому виводиться назва додатка, кнопка меню, що надає доступ до екрану вибору «розкладу» з головного екрану та кнопка для створення нових «розкладів» на екрані вибору «розкладу». Меню `ActionBar` реалізовано за допомогою метода `onCreateOptionsMenu()`, що підключає `xml`-файл, в якому вказані елементи меню. На кожній активності реалізоване своє меню, а тому у них підключаються різні `xml`-файли. Розглянемо структуру таких `xml`-файлів, які складаються з таких елементів:

- `<menu>` - кореневий елемент файлу, визначає меню у якому будуть знаходитись елементи `<group>`, `<item>`.
- `<item>` - безпосередньо пункт меню, тут можна вказати текст пункту меню, або вказати його іконку, також тут вказується режим відображення. Даний елемент може містити в собі також елемент `<menu>` що дозволяє створювати вкладені пункти меню.
- `<group>` - призначений для групування пунктів меню, це дозволяє здійснювати дії на певною групою меню відразу, а не над кожним елементом по черзі.

У даному випадку, у наявний всього лише один елемент натискання, який обробляється методом `onOptionsItemSelected()`.

Споріднені праці. В останні роки з ростом кількості мобільних пристроїв та додатків зростає і кількість прикладних програм для них. Оглянемо деякі споріднені з нашою розробкою додатки згідно наведених літературних даних.

Програмне забезпечення, представлене в роботі [12], для вибору режиму тренування на базі ОС Android. В даній роботі виконується розробка додатку засобами Android Studio, в якій представлено схожий підхід за допомогою фрагментів до реалізації вибору режиму тренування. Однак в ньому використовується старіша версія середовища розробки, що може мати негативний вплив на роботу програмного продукту в майбутньому. Додаток представляє собою базу даних, наповнену фізкультурними вправами. В поєднанні з даним додатком можна скласти вигідний для себе режим фізичного тренування.

Дослідження технології створення прикладних аплікацій для ОС ANDROID описане в роботі [13]. В даній праці показаний загальний принцип створення подібних додатків для Android та зроблено огляд засобів їх створення.

Висновки та перспективи подальшої роботи. Оглянувши статистику використання операційних систем на мобільних пристроях було обрано найбільш поширену систему для створення інтервального таймера. Оглянуто фрагментацію версій операційної системи та обрано найбільш поширений їх сегмент в поєднанні з найбільш поширеними видами екранів.

Реалізована задача розробки мобільного додатку інтервального таймера для операційної системи Android. На основі оглянутих існуючих літературних альтернатив додано новий функціонал для додатка, який дозволяє більш гнучко керувати інтервальними таймерами в цілому. Разом з перевіркою принципів роботи додатка з базою даних заодно перевірено і методи роботи з файлами ресурсів, а саме файлами © Мельник В.М., Вознюк О.В

теми, меню. Практика перевірки довела нормальне і коректне функціонування створеного інтервального таймера в поєднанні з базою даних та всіма видами необхідних додатку ресурсів.

Варто відмітити що даний прототип потрібно розвивати та, по можливості, оновлювати. Щодо подальшого вдосконалення розробки, її багатогранності та функціонального наповнення, в майбутньому можна реалізувати і деякі ідеї:

- збільшити кількість екранів підтримки в роботі;
- створити різновиди довідки для додатку;
- забезпечити відображення прогресу на екрані блокування;
- здійснити впровадження можливості відкладеного запуску;
- наповнити додаток новими функціями та можливостями Android.

1. Android [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Android>.
2. Smartphone OS Market Share [Електронний ресурс] – Режим доступу до ресурсу: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.
3. High-intensity interval training [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/High-intensity_interval_training.
4. Interval training [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Interval_training.
5. Android Studio 2.0 [Електронний ресурс] – Режим доступу до ресурсу: <http://android-developers.blogspot.com/2016/04/android-studio-2-0.html>.
6. Методика занятій оздоровительным бегом [Електронний ресурс] – Режим доступу до ресурсу: http://sportwiki.to/Методика_занятий_оздоровительным_бегом.
7. Dashboards [Електронний ресурс] – Режим доступу до ресурсу: <http://developer.android.com/intl/ru/about/dashboards/index.html>.
8. Манифест приложения [Електронний ресурс] – Режим доступу до ресурсу: <http://developer.android.com/intl/ru/guide/topics/manifest/manifest-intro.html>.
9. CountdownTimer [Електронний ресурс] – Режим доступу до ресурсу: <http://developer.android.com/intl/ru/reference/android/os/CountDownTimer.html>.
10. Хранение данных. SQLite [Електронний ресурс] – Режим доступу до ресурсу: <http://startandroid.ru/ru/uroki/vse-uroki-spiskom/74-urok-34-hranenie-dannyh-sqlite.html>.
11. SQLite на Android [Електронний ресурс] – Режим доступу до ресурсу: <http://developer.alexanderklimov.ru/android/sqlite/android-sqlite.php>.
12. Здолбіцька Н. В. Програмне забезпечення для вибору режиму тренування на базі ОС Android / Н. В. Здолбіцька, А. П. Здолбіцький, Ю. Ю. Найдюк // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2015. – № 18. – С. 61-65. – Режим доступу: http://nbuv.gov.ua/UJRN/Kitonv_2015_18_11.
13. Махун Д. Дослідження технології створення прикладних аплікацій для ОС ANDROID / Д. Махун, Б. Демида // Вісник Національного університету "Львівська політехніка". Комп'ютерні науки та інформаційні технології. – 2013. – № 771. – С. 111-120. – Режим доступу: http://nbuv.gov.ua/UJRN/VNULPKNIT_2013_771_19.