

УДК 681.515.8

Здолбіцька Н.В. к.т.н. доц., Здолбіцький А.П., Найдюк Ю.Ю.

Луцький національний технічний університет

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИБОРУ РЕЖИМУ ТРЕНУВАННЯ НА БАЗІ ОС ANDROID

**Здолбіцька Н.В., Здолбіцький А.П., Найдюк Ю.Ю. Програмне забезпечення для вибору режиму тренування на базі ОС Android.** Розроблено програмне забезпечення для можливості підбору оптимального та найкращого режиму занять в тренажерному залі, яке працюватиме на ОС Android. Мобільний додаток може служити посібником, довідником та помічником для великої кількості людей, які займаються спортом, слідкують та покращують власне здоров'я.

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

**Рис. 3. Літ. 12.**

**Здолбицкая Н.В., Здолбицкий А.П., Найдюк Ю.Ю. Программное обеспечение для выбора режима тренировки на базе ОС Android.** Разработано программное обеспечение для возможности подбора оптимального и лучшего режима занятий в тренажерном зале, которое будет работать на ОС Android. Мобильное приложение может служить пособием, справочником и помощником для большого количества людей, которые занимаются спортом, следят и улучшают свое здоровье.

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

**Zdolbitska N., Zdolbitsky A., Naidiyk Y. Software training mode selection based on Android OS.** Software to enable selection of the optimal and best treatment sessions in the gym, which will run on the operating system Android are developed. Mobile application can serve as a guide and assistant guide for a large number of people involved in sports, follow and improve their health.

**Keywords:** android, software, database, application, activity, sqlite, xml.

**Постановка проблеми.** В ході роботи була поставлена задача створення додатка довідника для мобільних пристроїв на операційній системі Android. Основною ідеєю була реалізація можливості вибору індивідуальної програми режиму занять в тренажерному залі.

Мовою програмування Android є Java. Для розробки використовується середовище розробки Android Studio, яке є надбудовою AndroidSDK і вбудованим в SDK емулятором мобільного приладу. Для створення та редагування растрових зображень використовувався Adobe Photoshop з дизайнерського пакету Creative Suite.

Також в процесі розробки документації були використані Enterprise Architect та Microsoft Office.

**Метою роботи** є створення програмного забезпечення для вибору режиму тренування, вивчення принципів роботи операційної системи Android, а також вивчення основних можливостей та функцій системи.

Однією з головних проблем розробки додатків на платформі Android є велика фрагментація приладів з різним розширенням екрана мів та версіями операційної системи.

Фрагментація девайсів збільшується, і разом з цим росте вибір Android. Виробники більш дешевих пристроїв намагатимуться використовувати останні версії Android, а фрагментація сприятиме екосистемі ставати більш глобальною і соціально-економічною.

Враховуючи статистику фрагментацій екранів та версії ОС (рис. 1), виберемо оптимальний пристрій для розроблення додатку.

Фрагментація географічно розширює ринок Android – вона не обмежується лише розсіюванням пристроїв і ОС. Вона також важлива для розуміння, які пристрої зараз актуальні, які девайси можуть бути актуальні для одного регіону, а для іншого навпаки, розробникам потрібно враховувати відмінності в продуктивності мережі в різних регіонах при розробці додатків. Інший фактор – це термін життя батареї; якщо сьогодні термін життя батареї цілком прийнятний, то в майбутньому її може не вистачати.

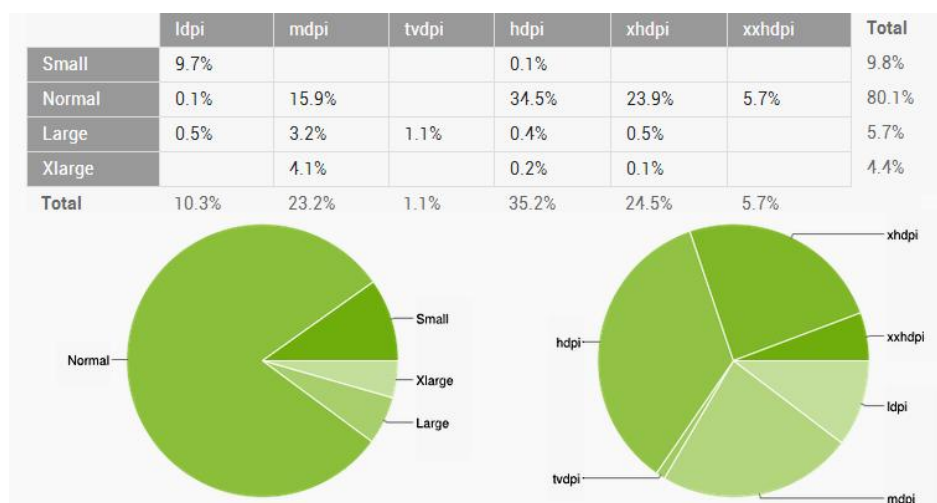


Рис. 1 – Фрагментація екранів

Згідно статистичним даним, при створенні додатку вибрано версію компіляції коду – Jelly Bean 4.1 API16 (37,3%) (мінімальна версія Gingerbread 2.3.3 API 10 (36,3%)). Розробка велася на пристрої з розміром hdpi normal (34,5%), а саме Motorola MB526 та AVD емулятор.

В даній статті створено додаток для вибору програми тренувань в тренажерному залі. Додаток містить довідник вправ, інформацію про калорійність їжі та спортивне харчування, тест вибору оптимальних вправ користувача та їх збереження в файл, секундомір з можливістю послідовного фіксування часу, форму підрахунку ідеальної маси тіла, синхронізацію з тематичними веб-ресурсами та можливості зворотнього зв'язку з розробником.

**AndroidManifest.xml.** Файл маніфесту AndroidManifest.xml надає основну інформацію системі про програму. Кожна програма повинна мати свій файл AndroidManifest.xml. Редагувати файл маніфесту можна вручну, змінюючи XML-код або через візуальний редактор Manifest Editor (Редактор файлу маніфесту), який дозволяє здійснювати візуальне і текстове редагування файлу маніфесту додатки.

Призначення файлу:

- оголошує ім'я Java-пакета додатка, який є унікальним ідентифікатором;
- описує компоненти програми – діяльності, служби, приймачі широкомовних намірів і контент-провайтери, що дозволяє викликати класи, які реалізують кожний з компонентів;
- містить список необхідних дозволів для звернення до захищених частин API та взаємодії з іншими додатками;
- оголошує дозволи, які сторонні додатки зобов'язані мати для взаємодії з компонентами цього додатка;
- оголошує мінімальний рівень API Android, необхідний для роботи програми;
- перераховує пов'язані бібліотеки;

Файл маніфесту інкапсулює всю архітектуру Android додатка, його функціональні можливості і конфігурацію. У процесі розробки програми файл постійно редагується, змінюється його структура, доповнюючись новими елементами і атрибутами.

Кореневим елементом маніфесту є <manifest>. Крім даного елемента обов'язковими елементами є теги <application> і <uses-sdk>. Елемент <application> є основним елементом маніфесту і містить безліч дочірніх елементів, що визначають структуру і роботу програми. Порядок розташування елементів, що знаходяться на одному рівні, довільний. Всі значення встановлюються через атрибути елементів. Крім обов'язкових елементів, згаданих вище, в маніфесті по мірі необхідності використовуються інші елементи.

Початок файлу, а саме налаштування додатку, має вигляд:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.naidiuk.mp" >
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

```
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.VIBRATE" />
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    android:orientation="vertical" >
    ...
```

**База даних SQLite.** SQLite доступний на будь-якому Android-пристрої, його не потрібно встановлювати окремо.

SQLite підтримує типи TEXT (аналог String в Java), INTEGER (аналог long в Java) і REAL (аналог double в Java). Решта типів слід конвертувати, перш ніж зберігати в базі даних. SQLite сама по собі не перевіряє типи даних, тому можна записати ціле число в колонку, призначену для рядків і навпаки.

Для нашого додатку потрібна досить проста база даних, яка міститиме лише колонку name, у якій будуть вказані назви вправ. Дані вправи виймаються з бази по результатам тесту, який проходить користувач.

Для обробки описаних вище ситуацій нам треба створити клас, який є спадкоємцем для SQLiteOpenHelper. Назвемо його sqllite. Цей клас надасть нам методи для створення або поновлення БД у випадках її відсутності або старіння.

При реалізації цього допоміжного класу ховається логіка, на основі якої приймається рішення про створення або оновлення бази даних перед її відкриттям. Клас SQLiteOpenHelper містить два абстрактних методи:

onCreate() – метод, який буде викликаний, якщо БД, до якої ми хочемо підключитися – не існує.

onUpgrade() – буде викликаний в разі, якщо ми намагаємося підключитися до БД більш нової версії, ніж існуюча.

**Файл стилів styles.xml.** Щоб наш додаток був якісним, можна скористатися спеціальною темою. Тема – це колекції стилів, які забезпечують професійний вид додатка, щоб він був схожий на рідний додаток Android. Сама система Android вже має кілька попередньо встановлених тем, якими можна скористатися в своїх цілях. Для цього достатньо тільки вказати ім'я теми в маніфесті.

Щоб полегшити громіздку роботу над створенням стилів інтерфейсу, було використано веб-ресурс Android Action Bar Style Generator. Так як вид Activity при створенні проекту був вибраний з Action Bar.

Даний ресурс створює набір графічних 9patch елементів та файли стилів, в тому числі style.xml, яким потрібно замінити стандартний.

У генераторі є можливість вибору таких властивостей:

- Style name – назва стилю;
- Style compatibility – сумісність стилю (Holo, App Compat, Sherlock);
- Base theme – базова тема (світла, темна, світла з темним Action Bar);
- Action bar style – стиль Action Bar (монотонний, прозорий);
- Action bar texture – текстурованість Action bar;
- Tab hairline style – лінії відмежування для табів;
- Neutral pressed states – звичайні натиснення (стандартні для версії KitKat);
- Action bar color – колір Action bar;
- Stacked color – колір табів;
- Tab indicator color – колір індикатора табів;
- Popup color – колір меню;
- Accent color – колір вибраного елемента;
- Action mode background color – колір заднього фону;
- Action mode highlight color – колір переднього фону.

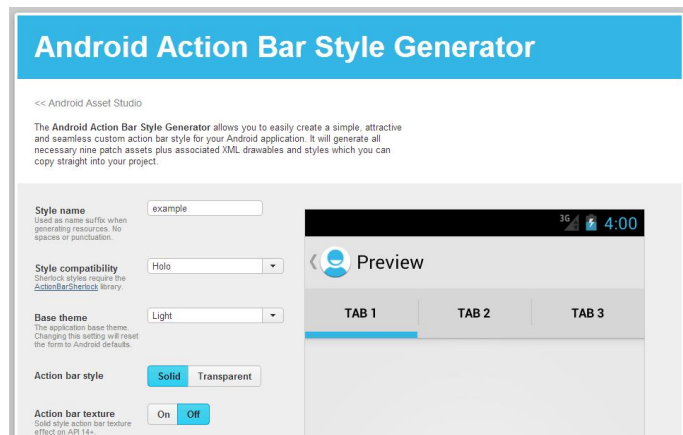


Рис. 2 – Вікно генератора стилів

**Фрагменти в Android.** Існує два основних підходи у використанні фрагментів.

Перший спосіб заснований на заміщенні батьківського контейнера. Створюється стандартна розмітка і в тому місці, де будуть використовуватися фрагменти, розміщується контейнер, наприклад, `LinearLayout`. У коді контейнер заміщається фрагментом. При використанні подібного сценарію в розмітці не використовується тег фрагмента, оскільки його не можна міняти динамічно. Також вам доведеться оновлювати `ActionBar`, якщо він залежить від фрагмента.

Другий варіант – використовуються окремі розмітки для телефонів і планшетів, які можна розмістити в різних папках ресурсів. Наприклад, якщо в планшеті використовується двохпанельна розмітка з двома фрагментами на одній активності, тоді використовується ця ж активність для телефону, але з підключенням іншої розмітки, яка містить один фрагмент. Для перемикання на другий фрагмент запускається друга `Activity`.

У додатку застосовано перший варіант, для відображення кількох табів з контейнерами поверх одного `Activity`.

Клас фрагменту подібно класу активності посилається на `xml` файл інтерфейсу, але має іншу конструкцію в якій метод `onCreate()` замінюється на `onCreateView()`.

**Меню додатка.** За меню відповідає клас `android.view.Menu`. Кожна активність пов'язана з одним об'єктом меню. Саме меню містить пункти меню (клас `android.view.MenuItem`) і підменю (клас `android.view.SubMenu`).

При натисканні кнопки меню на старіших моделях телефонів з'являється набір пунктів меню, що прикріплюється до активності. Меню може містити іконки. Таке меню може містити шість пунктів (як правило). При наявності більше шести пунктів використовується розширене меню – в цьому випадку замість шостого пункту з'являється пункт Параметри (Докладніше). При натисканні даного пункту показується розширене меню зі списком пунктів, які не вмістилися в основній частині меню вибору опцій.

Коли меню відкривається вперше, `Android` викликає метод `onCreateOptionsMenu()`, передаючи як параметр об'єкт меню. Меню можна створювати у вигляді ресурсів в `xml` файлі або використовувати метод `add()`.

У стандартному проекті при виборі звичайного шаблону вже є пункт для меню з одного пункту «Налаштування» і виклик методу для меню.

Розглянемо роботу меню через ресурси. Для створення меню використовуються ресурси, які повинні зберігатися в `xml` файлі. Меню складається з наступних елементів:

– `<menu>` – визначає меню, яке буде містити пункти меню. Елемент `<menu>` повинен бути кореневим елементом в структурі файлу і може містити один або декілька елементів `<item>` і `<group>`;

– `<item>` – створює безпосередньо пункти меню. Даний елемент може мати вкладений елемент `<menu>` для створення підменю;

– `<group>` – при бажанні можна використовувати невидимий контейнер для елементів `<item>`. Це дозволяє досягти деяких ефектів.

Меню додатку, що розробляється, може викликатися лише у MainActivity.java, в якому описано три метода для його керування: onCreateOptionsMenu(), onPrepareOptionsMenu(), onOptionsItemSelected().

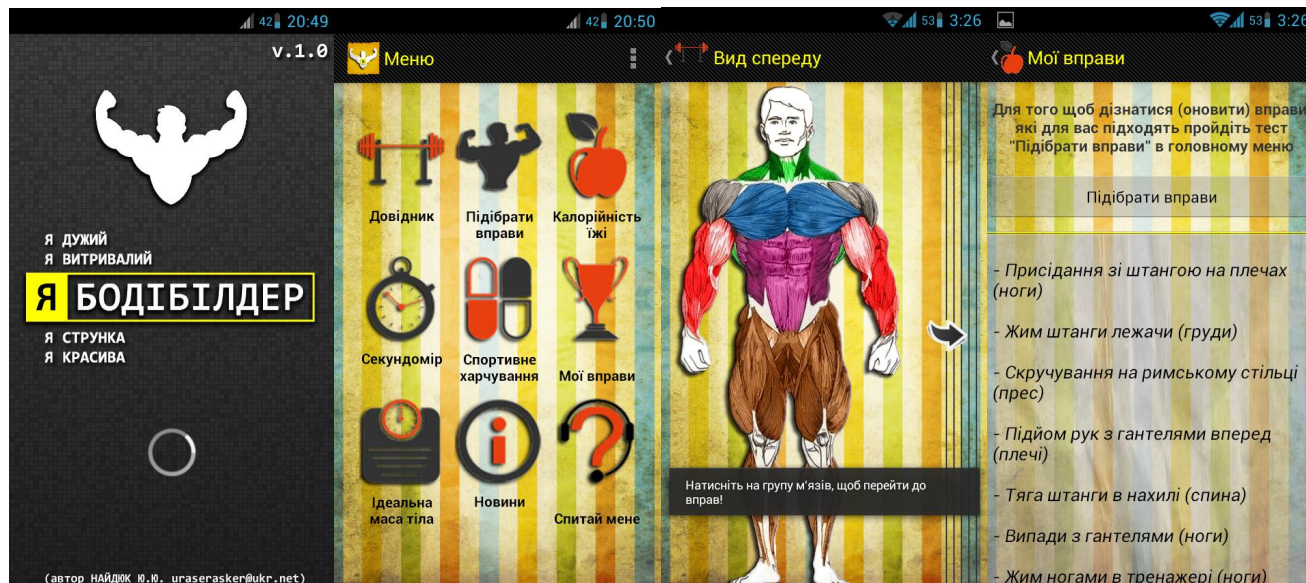


Рис. 3 – Інтерфейс додатка

### Висновки

Оглянувши всі найбільш поширені мобільні операційні системи було обрано ту, яка найбільш поширена, найбільш динамічно розвивається, має найбільший попит на ринку девайсів та найбільш відкрита для сторонніх розробників та програмістів.

Була поставлена задача створення додатка довідника для мобільних пристроїв на операційній системі Android. Основною ідеєю була реалізація можливості вибору індивідуальної програми режиму занять в тренажерному залі.

Додаток представляє собою «електронний довідник» спортсмена, бодібілдера, використовуючи який він може покращити свої показники в спорті, підтримувати задовільний стан здоров'я та вигляд тіла.

Слід відмітити, що даний прототип потрібно розвивати та оновлювати. Перечислимо деякі ідеї для подальшого удосконалення системи:

- врахування в подальшому зміни фрагментації девайсів;
- можливість редагування та самостійного підбору програм та вправ;
- розширення бази даних занять;
- створення онлайн справки;
- введення статистичних вимірів та порівняльних таблиць результатів занять;
- впровадження нових функцій та можливосте й ОС Android.

1. Rumbaugh J. Getting Started – Using Use Cases to Capture Requirements. // Journal of Object Oriented Programming. - 1994. Vol. 7, № 5. P. 8–12.
2. Блэйк Мик Программирование под Android. - СПб.: Санкт-Петербург, 2012. - 496 с.
3. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд. - М.: ДМК Пресс, 2007.
4. Гленфорд Майерс, Том Баджетт, Кори Сандлер Искусство тестирования программ, 3-е издание = The Art of Software Testing, 3rd Edition. - М.: "Диалектика", 2012. - 272 с.
5. Голощапов А.Л. Google Android: Программирование для мобильных устройств. - СПб.: БХВ-Петербург, 2011.
6. Эйблсон Ф. Разработка приложений для платформы Android в среде Eclipse.
7. Жвалевский Андрей Смартфоны Android без напряжения. Руководство пользователя. - СПб.: Санкт-Петербург, 2012. - 224 с.
8. Рето Майер Программирование приложений для планшетных компьютеров и смартфонов. - СПб.: Санкт-Петербург, 2011. - 672 с.
9. Хашими С., Коматинени С., Маклин Д. Разработка приложений
10. <http://developer.android.com/index.html>
11. <http://startandroid.ru/ru/>
12. <http://developer.alexanderklimov.ru/android/>