

**Android 6 dla programistów : techniki tworzenia aplikacji / Paul Deitel,  
Harvey Deitel, Alexander Wald. – Gliwice, cop. 2016**

Spis treści

<b>Przedmowa</b>	<b>17</b>
<b>Zanim zaczniesz</b>	<b>27</b>
<b>Rozdział 1. Android – wprowadzenie</b>	<b>33</b>
1.1. Wstęp	34
1.2. Android – najpopularniejszy mobilny system operacyjny na świecie	34
1.3. Cechy systemu Android	35
1.4. System operacyjny Android	38
1.4.1. Android 2.2 (Froyo)	39
1.4.2. Android 2.3 (Gingerbread)	39
1.4.3. Android 3.0 - 3.2 (Honeycomb)	40
1.4.4. Android 4.0 - 4.0.4 (Ice Cream Sandwich)	40
1.4.5. Android 4.1 - 4.3 (Jelly Bean)	42
1.4.6. Android 4.4 (KitKat)	43
1.4.7. Android 5.0 i 5.1 (Lollipop)	43
1.4.8. Android 6 (Marshmallow)	45
1.5. Pobieranie aplikacji z serwisu Google Play	47
1.6. Pakiety	47
1.7. Zestaw narzędzi Android Software Development Kit (SDK)	50
1.8. Programowanie obiektowe – krótkie przypomnienie	53
1.8.1. Samochód jako obiekt	53
1.8.2. Metody i klasy	53
1.8.3. Konkretyzacja	53
1.8.4. Ponowne wykorzystanie	54
1.8.5. Komunikaty i wywołania metod	54
1.8.6. Atrybuty i zmienne egzemplarzowe	54
1.8.7. Hermetyzacja	54
1.8.8. Dziedziczenie	54
1.8.9. Analiza i projektowanie zorientowane obiektowo (OOAD)	55
1.9. Sprawdzanie działania aplikacji Tip Calculator za pomocą urządzenia wirtualnego z systemem Android (AVD)	55
1.9.1. Otwarcie projektu Tip Calculator w środowisku Android Studio	55
1.9.2. Tworzenie wirtualnych urządzeń systemu Android	57
1.9.3. Uruchamianie aplikacji Tip Calculator na emulowanym smartfonie Nexus 6	60
1.9.4. Uruchamianie aplikacji Tip Calculator na realnym urządzeniu wyposażonym w system Android	63

1.10. Tworzenie doskonałych aplikacji przeznaczonych dla systemu Android	64
1.11. Dokumentacja przydatna podczas pracy nad aplikacjami systemu Android	66
1.12. Podsumowanie	68
<b>Rozdział 2. Aplikacja powitalna</b>	<b>69</b>
<i>Rozpoczęcie pracy w środowisku Android Studio — wizualne projektowanie graficznego interfejsu użytkownika, szablony, dostępność i internacjonalizacja</i>	
2.1. Wstęp	70
2.2. Przegląd technologii	71
2.2.1. Android Studio	71
2.2.2. Linear Layout, TextView i Image View	71
2.2.3. Rozszerzalny język znaczników (XML)	71
2.2.4. Zasoby aplikacji	71
2.2.5. Dostępność	72
2.2.6. Internacjonalizacja	72
2.3. Tworzenie aplikacji	72
2.3.1. Uruchamianie środowiska Android Studio	72
2.3.2. Tworzenie nowego projektu	72
2.3.3. Tworzenie nowego projektu w oknie Create New Project	72
2.3.4. Wybór docelowych urządzeń	74
2.3.5. Dodaj aktywność	75
2.3.6. Personalizacja aktywności	77
2.4. Okno Android Studio	77
2.4.1. Okno Project	78
2.4.2. Okna edytora	79
2.4.3. Okno Component Tree	80
2.4.4. Pliki z zasobami aplikacji	80
2.4.5. Edytor rozkładu	80
2.4.6. Domyślny graficzny interfejs użytkownika	81
2.4.7. Kod XML domyślnego interfejsu użytkownika	81
2.5. Tworzenie graficznego interfejsu użytkownika aplikacji za pomocą edytora rozkładu	83
2.5.1. Dodawanie obrazu	83
2.5.2. Dodawanie ikony aplikacji	85
2.5.3. Zmiana rozkładu RelativeLayout na LinearLayout	86
2.5.4. Zmiana identyfikatora i orientacji rozkładu LinearLayout	87
2.5.5. Konfigurowanie właściwości id i text pola TextView	88
2.5.6. Konfiguracja parametru textSize pola TextView — skalowane piksele (Scaled Pixels) i piksele niezależne od gęstości (den sity independent pixel)	90
2.5.7. Definiowanie parametru textColor pola TextView	91
2.5.8. Definiowanie parametru gravity pola TextView	92

2.5.9. Definiowanie parametru layout:gravity pola TextView	93
2.5.10. Definiowanie parametru layout:weight pola TextView	94
2.5.11. Dodawanie pola obrazu ImageView	95
2.5.12. Podgląd projektu	97
2.6. Uruchamianie aplikacji Welcome	98
2.7. Ułatwienie dostępu do aplikacji	100
2.8. Internacjonalizacja aplikacji	101
2.8.1. Lokalizacja	101
2.8.2. Nazywanie folderów zlokalizowanych zasobów	101
2.8.3. Dodawanie przetłumaczonych łańcuchów do projektu aplikacji	102
2.8.4. Lokalizacja łańcuchów	102
2.8.5. Testowanie hiszpańskiej wersji językowej na urządzeniu wirtualnym	103
2.8.6. Testowanie aplikacji w języku hiszpańskim na urządzeniu	104
2.8.7. TalkBack i lokalizacja	105
2.8.8. Rzeczy, o których należy pamiętać podczas lokalizacji	105
2.8.9. Profesjonalne tłumaczenie	105
2.9. Podsumowanie	106
<b>Rozdział 3. Aplikacja obliczająca napiwek</b>	<b>107</b>
<i>Wprowadzenie elementów takich jak GridLayout, EditText, SeekBar, obsługa zdarzeń, NumberFormat, personalizacja motywu aplikacji i definiowanie funkcjonalności aplikacji za pomocą kodu Java</i>	
3.1. Wstęp	108
3.2. Testowanie działania aplikacji Tip Calculator	109
3.3. Zastosowane rozwiązania	110
3.3.1. Klasa Activity	110
3.3.2. Metody cyklu roboczego klasy Activity	110
3.3.3. Biblioteka AppCompat i klasa AppCompatActivity	111
3.3.4. GridLayout i organizowanie widoków	112
3.3.5. Tworzenie graficznego interfejsu użytkownika w oknach Layout Editor, Component Tree i Properties	112
3.3.6. Formatowanie liczb na wartości walutowe zależne od lokalizacji i na łańcuchy procentowe	113
3.3.7. Implementacja interfejsu TextWatcher obsługującego obiekt EditText i zmiany tekstu	113
3.3.8. Implementacja interfejsu OnSeekBarChangeListener do obsługi zmian położenia suwaka SeekBar	113
3.3.9. Motywy Material Themes	113
3.3.10. Wytyczne material design: parametr elevation i cienie	114
3.3.11. Wytyczne material design dotyczące kolorów	114
3.3.12. AndroidManifest.xml	115
3.3.13. Wyszukiwanie w oknie Properties	115
3.4. Budowa graficznego interfejsu użytkownika	115
3.4.1. GridLayout — wprowadzenie	115

3.4.2. Tworzenie projektu TipCalculator	117
3.4.3. Zmiana rozkładu na GridLayout	117
3.4.4. Dodawanie obiektów TextViews, EditText i SeekBar	117
3.4.5. Personalizacja widoków	120
3.5. Domyślny motyw i personalizacja kolorów motywu	123
3.5.1. Motyw parent	123
3.5.2. Personalizacja kolorów motywu	124
3.5.3. Wspólne właściwości widoku jako style	126
3.6. Tworzenie logiki aplikacji	127
3.6.1. Instrukcje package i import	128
3.6.2. MainActivity — podklasa AppCompatActivity	129
3.6.3. Zmienne klasy i zmienne obiektowe	129
3.6.4. Przedefiniowanie metody onCreate klasy Activity	130
3.6.5. Metoda calculate klasy MainActivity	133
3.6.6. Anonimowa klasa wewnętrzna implementująca interfejs OnSeekBarChangeListener	133
3.6.7. Anonimowa klasa wewnętrzna implementująca interfejs TextWatcher	134
3.7. AndroidManifest.xml	135
3.7.1. Element manifest	136
3.7.2. Element application	136
3.7.3. Element activity	137
3.7.4. Element intent-filter	137
3.8. Podsumowanie	139

## **Rozdział 4. Aplikacja Flag Quiz** **141**

*Fragmenty, menu, preferencje, jawne intencje, obiekt handler, AssetManager, animacje, obiekty animujące, obiekty Toast, listy stanów kolorów, rozkłady obsługujące wiele orientacji urządzenia, tworzenie logów zawierających komunikaty przydatne podczas debugowania*

4.1. Wstęp	142
4.2. Testowanie aplikacji Flag Quiz	143
4.2.1. Konfiguracja quizu	143
4.2.2. Rozwiązywanie quizu	145
4.3. Omówienie technologii	148
4.3.1. Menu	148
4.3.2. Fragmenty	148
4.3.3. Metody cyklu życiowego fragmentu	149
4.3.4. Zarządzanie fragmentami	149
4.3.5. Preferencje	150
4.3.6. Folder assets	150
4.3.7. Foldery zasobów	150
4.3.8. Obsługa ekranów o różnych rozmiarach i rozdzielczościach	151
4.3.9. Określanie orientacji urządzenia	152
4.3.10. Wyświetlanie komunikatów za pomocą obiektów Toast	152

4.3.11. Korzystanie z klasy Handler w celu wykonania w przyszłości obiektu Runnable	152
4.3.12. Obiekty View i animacje	153
4.3.13. Korzystanie z klasy ViewAnimationUtils w celu stworzenia animacji okręgu odsłaniającego kolejny ekran	153
4.3.14. Określanie kolorów na podstawie stanu obiektu View za pośrednictwem listy kolorów stanów	153
4.3.15. Okno AlertDialog	153
4.3.16. Zapisywanie komunikatów wyjątków w pliku dziennika	154
4.3.17. Uruchamianie kolejnego obiektu Activity za pomocą jawnego komunikatu Intent	154
4.3.18. Java — struktury danych	155
4.3.19. Funkcje środowiska Java SE 7	155
4.3.20. AndroidManifest.xml	156
4.4. Tworzenie projektu, plików zasobów i dodatkowych klas	156
4.4.1. Tworzenie projektu	156
4.4.2. Rozkłady szablonu Blank Activity	157
4.4.3. Konfiguracja obsługi środowiska Java SE 7	158
4.4.4. Dodawanie obrazów flag do projektu	158
4.4.5. Plik strings.xml i sformatowane łańcuchy	158
4.4.6. arrays.xml	159
4.4.7. colors.xml	161
4.4.8. button_text_color.xml	162
4.4.9. Edycja pliku menu_main.xml	162
4.4.10. Tworzenie animacji trzęsącej się flagi	163
4.4.11. Definiowanie preferencji aplikacji za pomocą pliku preferences.xml	164
4.4.12. Dodawanie do projektu klas SettingsActivity i SettingsActivityFragment	166
4.5. Tworzenie graficznego interfejsu użytkownika aplikacji	167
4.5.1. Rozkład activity_main.xml dla urządzeń w orientacji pionowej	167
4.5.2. Projektowanie rozkładu fragment_main.xml	167
4.5.3. Pasek narzędzi graficznego edytora rozkładu	172
4.5.4. Rozkład content_main.xml i pozioma orientacja tabletu	173
4.6. Klasa MainActivity	175
4.6.1. Instrukcje package i import	175
4.6.2. Pola	176
4.6.3. Przedefiniowana metoda onCreate obiektu Activity	176
4.6.4. Przedefiniowana metoda onStart obiektu Activity	178
4.6.5. Przedefiniowana metoda onCreateOptionsMenu obiektu Activity	179
4.6.6. Przedefiniowana metoda onOptionsItemSelected obiektu Activity	180
4.6.7. Anonimowa klasa wewnętrzna implementująca obiekt OnSharedPreferenceChangeListener	180
4.7. Klasa MainActivityFragment	182
4.7.1. Instrukcje package i import	182

4.7.2. Pola	183
4.7.3. Przedefiniowywanie metody onCreateView obiektu Fragment	184
4.7.4. Metoda updateGuessRows	186
4.7.5. Metoda updateRegions	187
4.7.6. Metoda resetQuiz	187
4.7.7. Metoda loadNextFlag	189
4.7.8. Metoda getCountryName	191
4.7.9. Metoda animate	191
4.7.10. Anonimowa klasa wewnętrzna implementująca obiekt OnClickListener	193
4.7.11. Metoda disableButtons	195
4.8. Klasa SettingsActivity	196
4.9. Klasa SettingsActivityFragment	197
4.10. AndroidManifest.xml	197
4.11. Podsumowanie	199
<b>Rozdział 5. Aplikacja Doodlz</b>	<b>201</b>
<i>Grafika 2D, klasa Canvas, bitmapy, przyspieszeniometer, menedżer SensorManager, zdarzenia wielodotykowe, magazyn MediaStore, drukowanie, uprawnienia platformy Android 6.0, system Gradle</i>	
5.1. Wstęp	202
5.2. Testowanie działania aplikacji Doodlz za pomocą maszyny wirtualnej AVD	203
5.3. Technologie zastosowane w aplikacji	208
5.3.1. Metody cyklu życiowego obiektów Activity i Fragment	208
5.3.2. Personalizacja obiektów View	209
5.3.3. Korzystanie z obiektu SensorManager w celu nasłuchiwania zdarzeń przyspieszeniometera	209
5.3.4. Spersonalizowane obiekty DialogFragment	209
5.3.5. Rysowanie za pomocą klas Canvas i Paint, bitmapy	210
5.3.6. Przetwarzanie wielu zdarzeń dotknięcia ekranu i zapisywanie linii jako obiekty Path	210
5.3.7. Zapisywanie obrazów w pamięci urządzenia	211
5.3.8. Drukowanie i klasa PrintHelper biblioteki Android Support Library	211
5.3.9. Nowy model zezwoleń systemu Android 6.0 (Marshmallow)	211
5.3.10. Dodawanie zależności za pomocą systemu Gradle Build	211
5.4. Tworzenie projektu i zasobów	212
5.4.1. Tworzenie projektu	212
5.4.2. Gradle — dodawanie do projektu biblioteki pomocniczej	212
5.4.3. strings.xml	213
5.4.4. Importowanie ikon elementów znajdujących się w menu aplikacji	214
5.4.5. Menu MainActivityFragment	214
5.4.6. Dodawanie zezwoleń do pliku AndroidManifest.xml	216
5.5. Budowa graficznego interfejsu użytkownika aplikacji	217
5.5.1. Rozkład content_main.xml aktywności MainActivity	217

5.5.2. Rozkład fragment_main.xml fragmentu MainActivityFragment	217
5.5.3. Rozkład fragment_color.xml fragmentu ColorDialogFragment	218
5.5.4. Rozkład fragment_line_width.xml fragmentu LineWidthDialogFragment	219
5.5.5. Dodawanie klasy EraseImageDialogFragment	223
5.6. Klasa MainActivity	223
5.7. Klasa MainActivityFragment	224
5.7.1. Pola i instrukcje package i import	224
5.7.2. Przedefiniowana metoda onCreateView obiektu Fragment	226
5.7.3. Metody onResume i enableAccelerometerListening	226
5.7.4. Metody onPause i disableAccelerometerListening	227
5.7.5. Anonimowa klasa wewnętrzna przetwarzająca zdarzenia przyspieszeniomierza	228
5.7.6. Metoda confirmErase	229
5.7.7. Przedefiniowane metody obiektu Fragment: onCreateOptionsMenu i onOptionsItemSelected	230
5.7.8. Metoda saveImage	231
5.7.9. Przedefiniowana metoda onRequestPermissionsResult	233
5.7.10. Metody getDoodleView i setDialogOnScreen	233
5.8. Klasa DoodleView	234
5.8.1. Instrukcja package i instrukcje import	234
5.8.2. Zmienne statyczne i egzemplarzowe	235
5.8.3. Konstruktor	235
5.8.4. Przedefiniowana metoda onSizeChanged klasy View	236
5.8.5. Metody clear, setDrawingColor, getDrawingColor, setLineWidth i getLineWidth	237
5.8.6. Przedefiniowana metoda onDraw klasy View	238
5.8.7. Przedefiniowana metoda onTouchEvent klasy View	238
5.8.8. Metoda touchStarted	239
5.8.9. Metoda touchMoved	240
5.8.10. Metoda touchEnded	241
5.8.11. Metoda saveImage	242
5.8.12. Metoda printImage	243
5.9. Klasa ColorDialogFragment	243
5.9.1. Przedefiniowana metoda onCreateDialog klasy DialogFragment	244
5.9.2. Metoda getDoodleFragment	245
5.9.3. Przedefiniowane metody cyklu życia obiektu Fragment — onAttach i onDetach	246
5.9.4. Anonimowa klasa wewnętrzna, która reaguje na zdarzenia czterech suwaków SeekBar definiujących barwę	246
5.10. Klasa LineWidthDialogFragment	247
5.10.1. Metoda onCreateDialog	250
5.10.2. Anonimowa klasa wewnętrzna reagująca na zdarzenia paska widthSeekBar	250
5.11. Klasa EraseImageDialogFragment	250

5.12. Podsumowanie	252
<b>Rozdział 6. Aplikacja Cannon Game</b>	<b>253</b>
<i>Ręczna animacja klatka po klatce, grafika, dźwięk, tworzenie wielu wątków, obiekty SurfaceView i SurfaceHolder, tryb pełnoekranowy</i>	
6.1. Wstęp	254
6.2. Testowanie aplikacji Cannon Game	255
6.3. Zastosowane rozwiązania	255
6.3.1. Korzystanie z folderu zasobów res/raw	255
6.3.2. Metody cyklu życia klas Activity i Fragment	255
6.3.3. Przekazywanie metody onTouchEvent klasy View	256
6.3.4. Dodawanie dźwięku za pomocą narzędzi SoundPool i AudioManager	256
6.3.5. Tworzenie kolejnych klatek animacji za pomocą wątków i obiektów SurfaceView i SurfaceHolder	256
6.3.6. Proste wykrywanie zderzeń	257
6.3.7. Tryb pełnoekranowy	257
6.4. Tworzenie graficznego interfejsu użytkownika i plików zasobów	257
6.4.1. Tworzenie projektu	257
6.4.2. Modyfikacja motywu umożliwiająca usunięcie nazwy aplikacji i jej paska	258
6.4.3. strings.xml	258
6.4.4. Kolory	259
6.4.5. Dodawanie dźwięku	259
6.4.6. Dodawanie klasy MainActivityFragment	259
6.4.7. Edycja pliku activity_main.xml	259
6.4.8. Dodawanie obiektu Cannon View do fragmentu fragment_main.xml	260
6.5. Przegląd klas aplikacji	260
6.6. MainActivity — podklasa klasy Activity	261
6.7. MainActivityFragment — podklasa klasy Fragment	262
6.8. Klasa GameElement	263
6.8.1. Zmienne egzemplarzowe i konstruktor	264
6.8.2. Metody update, draw i playSound	265
6.9. Blocker — podklasa klasy GameElement	265
6.10. Target — podklasa klasy GameElement	266
6.11. Klasa Cannon	266
6.11.1. Zmienne egzemplarzowe i konstruktor	266
6.11.2. Metoda align	267
6.11.3. Metoda fireCannonball	268
6.11.4. Metoda draw	268
6.11.5. Metody getCannonball i removeCannonball	269
6.12. Cannonball — podklasa klasy GameElement	270
6.12.1. Zmienne egzemplarzowe i konstruktor	270
6.12.2. Metody getRadius, collidesWith, isOnScreen i reverseVelocityX	270
6.12.3. Metoda update	271



6.12.4. Metoda draw	272
6.13. Klasa Cannon View — podklasa SurfaceView	272
6.13.1. Instrukcje package i import	272
6.13.2. Stałe i zmienne egzemplarzowe	273
6.13.3. Konstruktor	274
6.13.4. Przedefiniowywanie metody onSizeChanged klasy View	276
6.13.5. Metody getWidth, getHeight i playSound	277
6.13.6. Metoda newGame	277
6.13.7. Metoda updatePositions	279
6.13.8. Metoda alignAndFireCannonball	280
6.13.9. Metoda showGameOverDialog	281
6.13.10. Metoda drawGameElements	282
6.13.11. Metoda testForCollisions	283
6.13.12. Metody stopGame i releaseResources	285
6.13.13. Implementacja metod interfejsu SurfaceHolder.Callback	285
6.13.14. Przedefiniowywanie metody onTouchEvent klasy View	286
6.13.15. Korzystanie z wątku CannonThread w celu utworzenia pętli gry	287
6.13.16. Metody hideSystemBars i showSystemBars	288
6.14. Podsumowanie	289

## **Rozdział 7. Aplikacja WeatherViewer** **291**

*Usługi sieciowe REST, zdarzenie asynchroniczne AsyncTask, połączenie HttpURLConnection, przetwarzanie danych JSON, JSONObject, JSONArray, ListView, ArrayAdapter, usługa ViewHolder*

7.1. Wstęp	292
7.2. Testowanie aplikacji WeatherViewer	293
7.3. Zastosowane rozwiązania	294
7.3.1. Usługi sieciowe	294
7.3.2. JavaScript Object Notation (JSON) i pakiet org.json	295
7.3.3. Połączenie HttpURLConnection i wywoływanie usługi sieciowej REST	297
7.3.4. Korzystanie z klasy AsyncTask w celu wygenerowania żądania sieciowego poza wątkiem graficznego interfejsu użytkownika	297
7.3.5. ListView, ArrayAdapter i wzorzec ViewHolder	298
7.3.6. Przycisk FloatingActionButton	299
7.3.7. TextInputLayout	299
7.3.8. Snackbar	300
7.4. Tworzenie graficznego interfejsu użytkownika aplikacji oraz plików zasobów	300
7.4.1. Tworzenie projektu	300
7.4.2. AndroidManifest.xml	300
7.4.3. strings.xml	301
7.4.4. colors.xml	301
7.4.5. activity_main.xml	301
7.4.6. content_main.xml	302
7.4.7. list_item.xml	302

7.5. Klasa Weather	304
7.5.1. Zmienne egzemplarzowe oraz instrukcje import i package	305
7.5.2. Konstruktor	305
7.5.3. Metoda convertTimeStampToDay	306
7.6. Klasa Weather Array Adapter	307
7.6.1. Instrukcja package i instrukcje import	307
7.6.2. Zagnieżdżona klasa ViewHolder	308
7.6.3. Zmienne egzemplarzowe i konstruktor	308
7.6.4. Przedefiniowana metoda get View adaptera ArrayAdapter	309
7.6.5. Podklasa AsyncTask służąca do pobierania obrazów w oddzielnym wątku	311
7.7. Klasa MainActivity	312
7.7.1. Instrukcja package i instrukcje import	312
7.7.2. Zmienne egzemplarzowe	313
7.7.3. Przedefiniowana metoda onCreate klasy Activity	314
7.7.4. Metody dismissKeyboard i createURL	315
7.7.5. Podklasa AsyncTask służąca do wywoływania usługi sieciowej	316
7.7.6. Metoda convert JSONtoArrayList	318
7.8. Podsumowanie	319

## **Rozdział 8. Aplikacja Twitter Searches** **321**

*Klasa SharedPreferences, podklasa SharedPreferences.Editor, niejawne intencje, wybór aktywności obsługującej intencję, klasa RecyclerView, podklasy RecyclerView.Adapter, RecyclerView.ViewHolder i RecyclerView.ItemDecoration*

8.1. Wstęp	322
8.2. Testowanie aplikacji Twitter Searches	323
8.2.1. Dodawanie ulubionych wyszukiwań	323
8.2.2. Przeglądanie wyników wyszukiwania wygenerowanych przez serwis Twitter	324
8.2.3. Edycja zapytania	326
8.2.4. Udostępnianie zapytania	327
8.2.5. Kasowanie zapytania	328
8.2.6. Przewijanie listy zapisanych zapytań	329
8.3. Zastosowane rozwiązania	329
8.3.1. Przechowywanie danych w postaci par klucz-wartość w pliku SharedPreferences	329
8.3.2. Niejawne intencje i wybór aplikacji obsługującej intencję	330
8.3.3. RecyclerView	331
8.3.4. Podklasy RecyclerView.Adapter i RecyclerView.ViewHolder	331
8.3.5. Podklasa RecyclerView.ItemDecoration	331
8.3.6. Wyświetlanie listy opcji w oknie AlertDialog	332
8.4. Tworzenie graficznego interfejsu użytkownika oraz plików zasobów	332
8.4.1. Tworzenie projektu	332
8.4.2. AndroidManifest.xml	332

8.4.3. Dodawanie biblioteki RecyclerView	332
8.4.4. colors.xml	333
8.4.5. strings.xml	333
8.4.6. arrays.xml	333
8.4.7. dimens.xml	334
8.4.8. Dodawanie ikony przycisku zapisywania	334
8.4.9. activity_main.xml	334
8.4.10. content_main.xml	335
8.4.11. Rozkład elementu RecyclerView — list_item.xml	337
8.5. Klasa MainActivity	338
8.5.1. Instrukcje package i import	338
8.5.2. Pola klasy MainActivity	339
8.5.3. Przedefiniowana metoda onCreate klasy Activity	340
8.5.4. Procedura obsługi zdarzeń TextWatcher i metoda updateSaveFAB	342
8.5.5. Interfejs OnClickListener nasłuchujący zdarzeń przycisku saveButton	343
8.5.6. Metoda addTaggedSearch	344
8.5.7. Anonimowa klasa wewnętrzna implementująca obiekt View.OnClickListener w celu wyświetlenia wyników wyszukiwania	345
8.5.8. Anonimowa klasa wewnętrzna, która implementuje interfejs View.OnLongClickListener umożliwiający udostępnienie, skasowanie i edycję zapisanego zapytania	346
8.5.9. Metoda shareSearch	348
8.5.10. Metoda deleteSearch	349
8.6. SearchesAdapter — podklasa RecyclerView.Adapter	350
8.6.1. Instrukcja package, instrukcje import, zmienne egzemplarzowe i konstruktor	350
8.6.2. Zagnieżdżona podklasa ViewHolder klasy RecyclerView.ViewHolder	351
8.6.3. Przedefiniowane metody klasy RecyclerView.Adapter	352
8.7. ItemDivider — podklasa klasy RecyclerView.ItemDecoration	353
8.8. Fabric — nowa platforma programistyczna serwisu Twitter	355
8.9. Podsumowanie	356

## **Rozdział 9. Aplikacja Address Book 357**

*Klasa FragmentTransaction i stos obiektów Fragment, SQLite, SQLiteDatabase, SQLiteOpenHelper, ContentProvider, ContentResolver, Loader, LoaderManager, Cursor i style graficznego interfejsu użytkownika*

9.1. Wstęp	358
9.2. Testowanie aplikacji Address Book	360
9.2.1. Dodawanie kontaktu	360
9.2.2. Przeglądanie informacji przypisanych do kontaktu	361
9.2.3. Edycja kontaktu	361
9.2.4. Kasowanie kontaktu	361
9.3. Zastosowane rozwiązania	362
9.3.1. Wyświetlanie obiektów Fragment za pomocą klasy	

FragmentTransaction	362
9.3.2. Przesyłanie danych pomiędzy fragmentem a nadrzędną aktywnością	363
9.3.3. Obsługa bazy danych SQLite	363
9.3.4. Obiekty ContentProvider i ContentResolver	363
9.3.5. Asynchroniczny dostęp do bazy danych — obiekt Loader i menedżer LoaderManager	364
9.3.6. Definiowanie stylów i przypisywanie ich do komponentów graficznego interfejsu użytkownika	365
9.3.7. Określanie tła pola TextView	365
9.4. Tworzenie graficznego interfejsu użytkownika oraz plików zasobów	365
9.4.1. Tworzenie projektu	365
9.4.2. Tworzenie klas	366
9.4.3. Dodawanie ikon aplikacji	367
9.4.4. strings.xml	367
9.4.5. styles.xml	367
9.4.6. textview_border.xml	369
9.4.7. Rozkład aktywności MainActivity	369
9.4.8. Rozkład fragmentu ContactsFragment	371
9.4.9. Rozkład fragmentu DetailFragment	372
9.4.10. Rozkład fragmentu AddEditFragment	373
9.4.11. Menu fragmentu DetailFragment	375
9.5. Przegląd klas opisanych w dalszej części tego rozdziału	376
9.6. Klasa DatabaseDescription	377
9.6.1. Pola static	377
9.6.2. Zagnieżdżona klasa Contact	378
9.7. Klasa AddressBookDatabaseHelper	379
9.8. Klasa AddressBookContentProvider	380
9.8.1. Pola klasy AddressBookContentProvider	381
9.8.2. Przedefiniowane metody onCreate i getType	382
9.8.3. Przedefiniowana metoda query	383
9.8.4. Przedefiniowana metoda insert	385
9.8.5. Przedefiniowana metoda update	387
9.8.6. Przedefiniowana metoda delete	388
9.9. Klasa MainActivity	389
9.9.1. Klasa nadrzędna, zaimplementowane interfejsy i pola	389
9.9.2. Przedefiniowana metoda onCreate	390
9.9.3. Metody interfejsu ContactsFragment.ContactsFragmentListener	391
9.9.4. Metoda displayContact	392
9.9.5. Metoda displayAddEditFragment	393
9.9.6. Metody interfejsu DetailFragment.DetailFragmentListener	394
9.9.7. Metody interfejsu AddEditFragment.AddEditFragmentListener	394
9.10. Klasa ContactsFragment	395
9.10.1. Klasa nadrzędna i implementowany interfejs	395
9.10.2. Interfejs ContactsFragmentListener	396

9.10.3. Pola	396
9.10.4. Przedefiniowana metoda onCreateView	397
9.10.5. Przedefiniowane metody onAttach i onDetach obiektu Fragment	398
9.10.6. Przedefiniowana metoda onCreateView obiektu Fragment	398
9.10.7. Metoda updateContactList	399
9.10.8. Metody interfejsu LoaderManager.LoaderCallbacks<Cursor>	399
9.11. Klasa Contacts Adapter	401
9.12. Klasa AddEditFragment	404
9.12.1. Klasa nadrzędna i implementowany interfejs	404
9.12.2. Interfejs AddEditFragmentListener	404
9.12.3. Pola	405
9.12.4. Przedefiniowane metody obiektu Fragment – onAttach, onDetach i onCreateView	406
9.12.5. Obiekt nasłuchujący zdarzeń TextWatcher nameChangeListener i metoda updateSaveButtonFAB	407
9.12.6. Metoda saveContact i obiekt saveContactButtonClicked nasłuchujący zdarzeń View.OnClickListener	408
9.12.7. Metody interfejsu LoaderManager.LoaderCallbacks<Cursor>	410
9.13. Klasa DetailFragment	411
9.13.1. Klasa nadrzędna i implementowany interfejs	411
9.13.2. Zagnieżdżony interfejs DetailFragmentListener	412
9.13.3. Pola	412
9.13.4. Przedefiniowane metody onAttach, onDetach i onCreateView	413
9.13.5. Przedefiniowane metody onCreateViewOptionsMenu i onOptionsItemSelected	414
9.13.6. Metoda deleteContact i fragment DialogFragment confirmDelete	415
9.13.7. Metody interfejsu LoaderManager.LoaderCallback<Cursor>	416
9.14. Podsumowanie	417
<b>Rozdział 10. Serwis Google Play i zagadnienia biznesowe związane z tworzeniem aplikacji</b>	<b>419</b>
10.1. Wstęp	420
10.2. Przygotowywanie aplikacji do publikacji	420
10.2.1. Testowanie aplikacji	421
10.2.2. Umowa pomiędzy licencjodawcą a użytkownikiem końcowym	421
10.2.3. Ikony i etykiety	421
10.2.4. Oznaczanie aplikacji numerem wersji	422
10.2.5. Tworzenie licencji określających zasady uzyskiwania dostępu do płatnych aplikacji	422
10.2.6. Zaciemnianie kodu	423
10.2.7. Uzyskiwanie prywatnego klucza przeznaczonego do cyfrowego podpisywania aplikacji	423
10.2.8. Obraz promocyjny i zrzuty ekranu	423
10.2.9. Materiał wideo promujący aplikację	425
10.3. Wycena aplikacji – darmowa czy płatna?	425

10.3.1. Płatne aplikacje	426
10.3.2. Darmowe aplikacje	426
10.4. Zarabianie na reklamach wyświetlanych w aplikacji	428
10.5. Zarabianie na sprzedaży wirtualnych dóbr za pomocą mechanizmu płatności wbudowanego w aplikację	428
10.6. Rejestracja w serwisie Google Play	430
10.7. Konfigurowanie konta sprzedawcy w serwisie Google Payments	430
10.8. Ładowanie aplikacji do serwisu Google Play	431
10.9. Otwieranie sklepu z aplikacjami z poziomu aplikacji	433
10.10. Zarządzanie aplikacjami udostępnionymi w serwisie Google Play	434
10.11. Inne serwisy pośredniczące w sprzedaży aplikacji systemu Android	434
10.12. Inne platformy mobilne i przenoszenie na nie aplikacji	434
10.13. Rozreklamowanie aplikacji	435
10.14. Podsumowanie	439
<b>Skorowidz</b>	<b>441</b>

oprac. BPK