Zadania bazują głównie na projektach z książki “Android. Programowanie aplikacji. Rusz głową!”.

https://zacniewski.github.io/teaching/2019-android

1. **Kalkulator**
   - bazuje na rozdziale nr 2;
   - wybieramy dwie liczby (każda zapisana na sztywno w listie rozwijanej typu Spinner) i po kliknięciu przycisku otrzymujemy wynik;
   - praca z zasobami łańcuchowymi i tablicowymi mile widziana;
   - należy użyć elementów typu Spinner (argumenty do działań - pierwsza liczba i druga liczba), Button (wybór działania, może być kilka Button’ów) i TextView (wyświetlanie wyniku).

2. **Aktywności i intencje**
   - bazuje na rozdziale nr 3;
   - tworzymy trzy aktywności;
   - za pomocą intencji tworzymy przejścia z każdej aktywności do pozostałych;
   - analiza pliku manifestu wskazana!
   - użycie metod putExtra() i getIntent() konieczne do uzyskania wyższej oceny;
   - użycie ‘akcji’ i filtrów intencji - do przetestowania.

3. **Widoki i grupy widoków w różnych układach**
   - bazuje na rozdziale nr 5 (5 i 6 w edycji nr 2);
   - należy utworzyć dwie aktywności - jedna z ConstraintLayout, druga z FrameLayout;
   - dla ConstraintLayout należy zaprojektować **własny** oryginalny układ elementów,
   - za pomocą intencji tworzymy przejścia z jednej aktywności do drugiej;
   - należy wykorzystać w wybranych aktywnosciach elementy takie jak:
     - przełącznik,
     - checkbox,
     - radio button,
     - lista rozwijana,
     - obrazek,
     - obrazek wyświetlony na przycisku,
     - widok przewijany,
     - komunikaty typu (Toast)
   - należy korzystać jak najwięcej z zasobów (np. łańcuchowych), gdzie powinny być umieszczone odpowiednie parametry layout’u.

4. **Widoki list, adaptery, paski aplikacji …**
   - bazuje na rozdziałach nr 6 i 9 (7 i 8 w edycji nr 2);
   - należy utworzyć aplikację co najmniej 3-poziomową (poziomy inne niż w ww. rozdziałach);
   - aktywność główna zawiera widok listy i dowolny obrazek (np. logo);
   - każda pozycja z listy po kliknięciu prowadzi do poziomu drugiego (aktywność kategorii), zawierającego kolejny widok listy;
   - każda pozycja z listy po kliknięciu prowadzi do poziomu trzeciego (aktywność szczegółów), zawierającego informacje o danym elemencie (produkcie itp.);
   - utwórz klasę przechowującą informacje o elementach list;
   - dodaj pliki graficzne do zasobów;
   - zaimplementuj obiekt nasłuchujący (reagowanie na kliknięcia widoku);
   - użyj adaptera do pobrania danych statycznych;
   - usuń pasek aplikacji, dodaj pasek narzędzi i zmień wybrany motyw, pozmieniaj również odpowiednie kolory;
   - dla paska narzędzi utwórz osobny layout, w aktywności należy skorzystać z import android.support.v7.widget.Toolbar;
   - ww. pasek narzędzi należy dodać do każdej utworzonej aktywności (znacznik ‘include’ w plikach layoutu);
   - pamiętaj o metodzie setSupportActionBar(); *

- utwórz aktywność InfoActivity, która powinna zawierać krótką informacje o aplikacji;
- dla każdej aktywności dodaj w pliku manifestu label, który będzie wyświetlany na pasku narzędziowym;
- w głównej aktywności dodaj akcję do paska narzędziowego (ikonka + tytuł akcji + plik zasobu menu);
- w głównej aktywności dodaj menu do paska aplikacji w metodzie onCreateOptionsMenu();
- w głównej aktywności dodaj reakcję na kliknięcia elementów akcji za pomocą metody onOptionsItemSelected();
- dla InfoActivity ustaw aktywność nadrzędną w pliku manifestu - wybierz aktywność główną;
- dla InfoActivity dodaj przycisk 'w górę' wykorzystując metody getSupportActionBar() i setDisplayHomeAsUpEnabled();

5. **Fragmenty**
   - bazuje na rozdziałach nr 7-8 (9-11 w drugiej edycji);
   - aplikacja powinna dotyczyć programowania (o ile ktoś nie kontynuuje apek z poprzednich laborek),
   - utwórz aktywność, która będzie korzystać z z co najmniej dwóch fragmentów,
   - pamiętaj o layoutach fragmentu, klasie Fragment, metodzie onCreateView() w plikach .java,
   - fragmenty mogą pobierać dane z pliku *.java lub z bazy danych,
   - pamiętaj o obiekcie klasy LayoutInflater - przygotowanie układu (ang. inflate) oznacza przekształcenie widoków określonych w kodzie XML układu na obiekty Javy,
   - skorzystaj z menedżera fragmentów, np. getFragmentManager(), w celu pobrania referencji do fragmentu,
   - wykorzystaj interfejs do komunkacji aktywności z fragmentem,
   - wykorzystaj transakcję fragmentu;
   - utwórz fragment zagnieżdżony w innym fragmencie,
   - pamiętaj o zagnieżdżonych transakcjach,
   - wykorzystaj interfejs OnClickListener do obsługi kliknięć.

6. **Baza danych SQLite i kursory**
   - bazuje na rozdziale nr 11 i 12 (15 i 16 w drugiej edycji);
   - należy rozwinąć lub zmodyfikować apki z innych zadań, tak by dane pobierane były z bazy danych zamiast z plików;
   - należy utworzyć bazę danych SQLite do przechowywania ww. danych;
   - skorzystaj z klasy SQLiteOpenHelper() do tworzenia i utrzymania ww. bazy,
   - pamiętaj o metodach onCreate() i onUpgrade(), ewentualnie onDowngrade(),
   - skorzystaj z obiektu typu Cursor() i metody query() do tworzenia zapytań,
   - wykorzystaj co najmniej dwa różne zapytania w aplikacji,
   - skorzystaj z wybranych metod poruszania się po kursorze: moveToFirst(), moveToLast(), moveToPrevious(), moveToNext().

7. **Zadania asynchroniczne**
   - bazuje na rozdziale nr 12 (17 w drugiej edycji);
   - przeanalizuj rodzaje wątków w aplikacjach na Androida,
   - wykorzystaj klasę AsyncTask do obsługi kodu, korzystającego z bazy danych, umieszczając go w osobnym wątku,
   - zaimplementuj metody onPreExecute(), doInBackground(), onPostExecute() oraz onProgressUpdate(),
   - pamiętaj o parametrach Params, Progress oraz Results klasy AsyncTask,
   - spróbuj oszacować różnicę czasów dla apki bez klasy AsyncTask i z użyciem tej klasy klasą. Skorzystaj np. z klasy TimingLogger.

8. **Usługi uruchomione i usługi powiązane**
    - bazuje na rozdziale nr 13 (nr 18 i 19 w drugiej edycji);
    - part I:
        - utwórz usługę uruchomioną korzystając z klasy IntentService, do innego celu niż w książce;
        - usługa powinna być uruchamiana po kliknięciu przycisku w aktywności,
        - pamiętaj o metodach onHandleIntent() i startService(),
        - Użyj usługi np. do wyświetlenia wybranego tekstu najpierw w dzienniku zdarzeń, a w drugim kroku jako powiadomienie lub do wybranego przez siebie zadania;
        - skorzystaj z obiektu typu NotificationManager.
    - part II:
        - utwórz usługę powiązaną, służącą do pomiaru odległości,
        - wykorzystaj systemową usługę lokalizacyjną,
        - utwórz aktywność, która będzie komunikować się z ww. usługą,
        - wykorystaj obiekt typu Binder do powiązania aktywności z usługą. Usługa powiązana tworzy obiekt Binder, zawiera on zawiera referencję do usługi powiązanej.
        - wykorzystaj obiekt typu ServiceConnection. Aktywność tworzy obiekt ServiceConnection, służy on do utworzenia połączenia z usługą.
        - zaimplementuj interfejs LocationListener i zarejestruj obiekt tego typu w systemowej usłudze lokalizacyjnej,
        - w momencie tworzenia usługi należy przygotować obiekt nasłuchujący, który będzie odbierać informacje o zmianach lokalizacji urządzenia
        - utwórz metodę do pomiaru odległości korzystając z metody onLocationChanged(Location() obiektu typu LocationListener,
        - pamiętaj o metodzie bindService(), aby powiązać aktywność z usługą,
        - pamiętaj o modyfikacji pliku manifestu, w celu uzyskania uprawnień do korzystania z odbiornika GPS,