

# TELEINFORMATYCZNE SYSTEMY MOBILNE

## DOKUMENTACJA PROJEKTU „TSM Calculator”

<https://github.com/jackmie/TSM-Calculator>

Data wykonania: 09.09.2020 r.

Prowadzący: dr Paweł Misiorek

Autor: Jacek Kmiecik

Informatyka, st. niestacjonarne, sem. 8

## Wprowadzenie

Celem niniejszego projektu było stworzenie aplikacji mobilnej na urządzenie typu smartfon obsługującej wyświetlanie reklam w jednym z dostępnych systemów RTB oraz oferującej użytkownikowi dowolną praktyczną funkcjonalność.

Aplikacja nazwana została „TSM Calculator” i oferuje funkcjonalność prostego kalkulatora dla dwóch liczb wejściowych typu całkowitego.

Temat ten został wybrany celem poznania podstaw tworzenia aplikacji mobilnych, jak również ich monetyzacji poprzez wbudowane reklamy Google AdMob.

## Wykorzystane technologie i narzędzia

### Java + Android SDK

Aplikacja została napisana z przeznaczeniem na urządzenia mobilne z systemem Android. W tym celu wykorzystany został język Java z Android SDK w wersji 30.

### Google AdMob

Do wyświetlania reklam w aplikacji wykorzystana została platforma Google AdMob. Aplikacja korzysta z upublicznionej konfiguracji testowej dla reklam AdMob.

### Android Studio

Aplikacja mobilna została stworzona w IDE Android Studio.

## Implementacja

Aplikację stanowi jedna główna aktywność (ang. *Activity*). W ramach tej aktywności utworzono:

- dwa obiekty EditText do wprowadzania danych wejściowych (liczb całkowitych),
- jeden obiekt TextView do wyświetlenia rezultatu wykonanych obliczeń,

- cztery obiekty typu Button jako przyciski służące do wykonania konkretnego działania,
- jeden obiekt AdView do wyświetlenia reklamy typu Baner.

```

<EditText
    android:id="@+id/number2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Number 2"
    android:ems="10"
    android:inputType="number" />

<TextView
    android:id="@+id/result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="25sp"
    android:text="Result" />

<Button
    android:id="@+id/add"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Add" />

```

Rys. 1 Dodawanie obiektów do aktywności.

```

<com.google.android.gms.ads.AdView
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    ads:adSize="BANNER"
    ads:adUnitId="ca-app-pub-3940256099942544/6300978111"></com.google.android.gms.ads.AdView>

```

Rys. 2 Dodawanie obiektu AdView do aktywności.

Dane do backendu pobierane są za pomocą metody *findViewById*, następnie rzutowane są na odpowiedni typ.

```
result = (TextView)findViewById(R.id.result);

number1 = (EditText)findViewById(R.id.number1);
number2 = (EditText)findViewById(R.id.number2);

add = (Button)findViewById(R.id.add);
subtract = (Button)findViewById(R.id.subtract);
multiply = (Button)findViewById(R.id.multiply);
divide = (Button)findViewById(R.id.divide);
```

Rys. 3 Pobieranie danych do backendu.

Operacje na danych wejściowych wykonywane są za pomocą funkcji obliczeniowych wywoływanych w momencie kliknięcia na odpowiedni przycisk. Wartości wejściowe pobierane są jako łańcuch tekstowy (metoda *toString*) a następnie parsowane do wartości całkowitych *int* lub zmiennoprzecinkowych *float* (dla operacji dzielenia). Następnie wykonywane jest działanie, którego wynik ustawiany jest jako wartość pola tekstowego *TextView*.

```
add.setOnClickListener((v) -> {
    num1 = Integer.parseInt(number1.getText().toString());
    num2 = Integer.parseInt(number2.getText().toString());
    res_value = num1 + num2;
    result.setText(String.valueOf(res_value));
});
```

Rys. 4 Operacja dodawania danych wejściowych.

Wyświetlanie reklam odbywa się w kilku krokach.

1. Jako obiekt typu `AdView` pobierany jest widok pola służącego do wyświetlenia reklamy.
2. Utworzone i zbudowane (*build*) jest zapytanie `AdRequest` zawierające dane użytkownika celem wyświetlania personalizowanych reklam (*targeted advertisements*).
3. Do danego widoku `AdView` ładowana jest reklama uzyskana na podstawie zapytania `AdRequest`.

```
AdView adView = (AdView)findViewById(R.id.adView);

AdRequest adRequest = new AdRequest.Builder()
    // .addTestDevice(AdRequest.DEVICE_ID_EMULATOR)
    .build();

adView.loadAd(adRequest);
```

Rys. 5 Wyświetlenie reklamy.

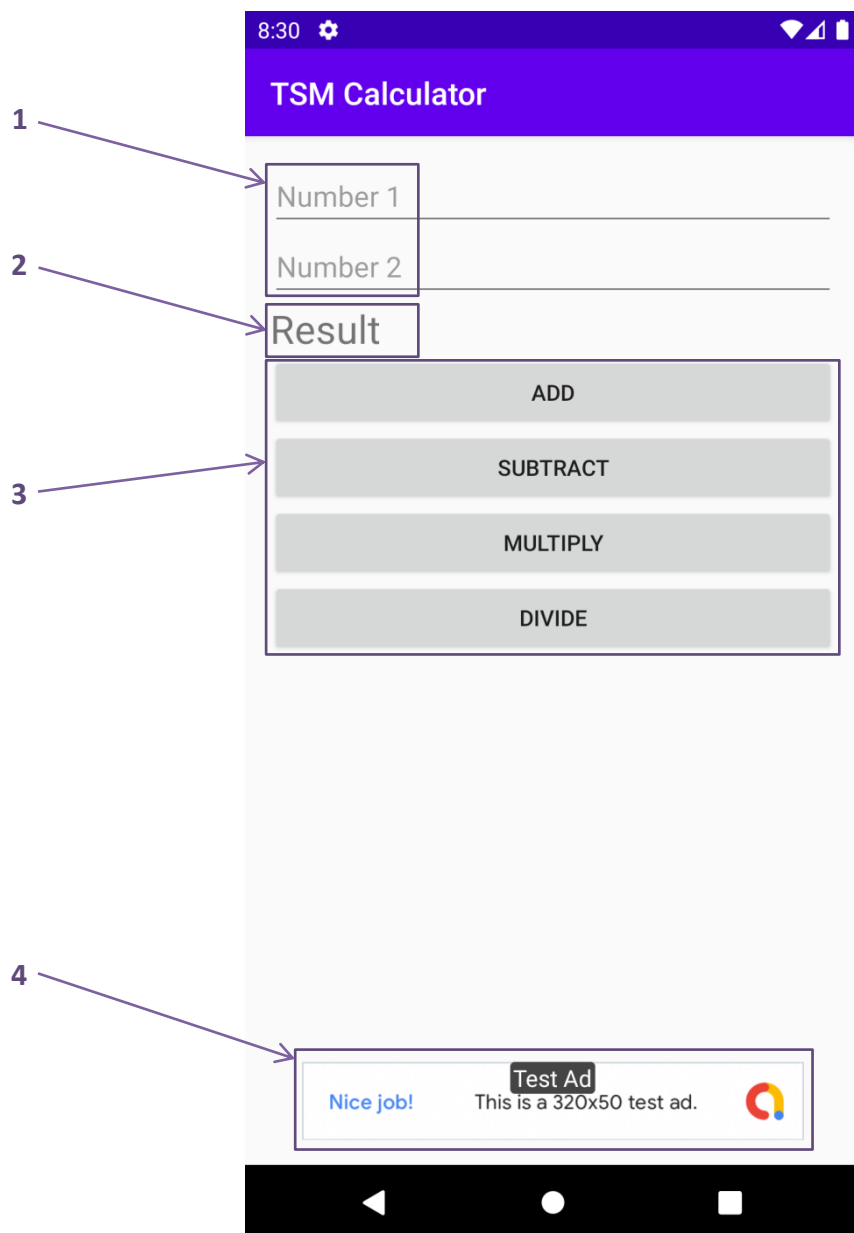
Aby otrzymać reklamy testowe podczas testowania dodano ID urządzenia testowego za pomocą funkcji *addTestDevice*.

## Dokumentacja użytkowa

Aplikację można zainstalować uruchamiając na telefonie z systemem Android plik *app-release.apk* dostępny tutaj:

<https://github.com/jackmie/TSM-Calculator/blob/master/app/release/app-release.apk>

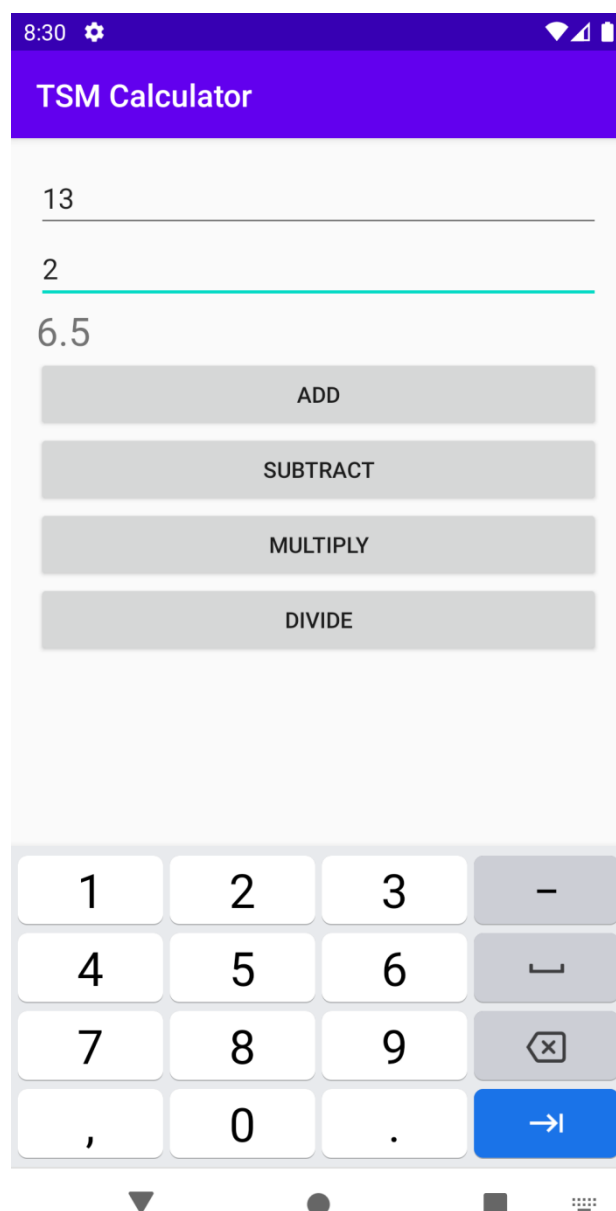
Aplikacja składa się z jednego widoku, na którym widoczne są dane opisane poniżej.



Rys. 6 Interfejs aplikacji.

1. Pola na dane wejściowe
2. Pole wyniku działania
3. Przyciski funkcyjne – wykonanie obliczeń
4. Baner reklamowy

Przykład wykonania operacji dzielenia w aplikacji:



Rys. 7 Przykładowe działanie aplikacji.