

Android

Systemy operacyjne na
platformach mobilnych

O czym
będziemy
mówić

Co to
jest?

Jak to
wygląda?

Czy to jest
praktyczne?

Zastosowanie

Dziękujemy
za
uwagę

View Models

Przywracanie stanu aplikacji po
obrocie lub minimalizacji ekranu



The
team

The team



Michał Niksa



Krzysztof
Gruszczyński



Adam Wanat

Android

Systemy operacyjne na
platformach mobilnych

O czym
będziemy
mówić

Co to
jest?

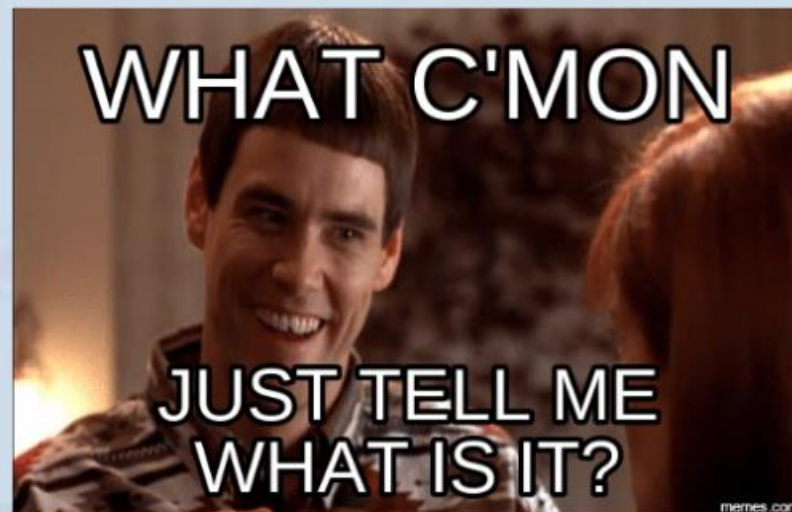
Jak to
wygląda?

Czy to jest
praktyczne?

Zastosowanie

Dziękujemy
za
uwagę

Co to jest?



ViewModel to jest klasa zaprojektowana do przechowywania i zarządzania wszelkiego rodzaju danymi związanymi z interfejsem graficznym.

Dzięki niej można również zachować wszelkie dane, które normalnie zostałyby utracone podczas zmian widoków czy ekranów

Tej klasy używa się do zarządzania danymi na klasach Activity oraz Fragment, i będzie działać tak długo, jak działa dana klasa

Dodatkowo klasa ViewModel może być użyta jako warstwa komunikacji pomiędzy aktywnościami i fragmentami

Android

Systemy operacyjne na
platformach mobilnych

O czym
będziemy
mówić

Co to
jest?

Jak to
wygląda?

Czy to jest
praktyczne?

Zastosowanie

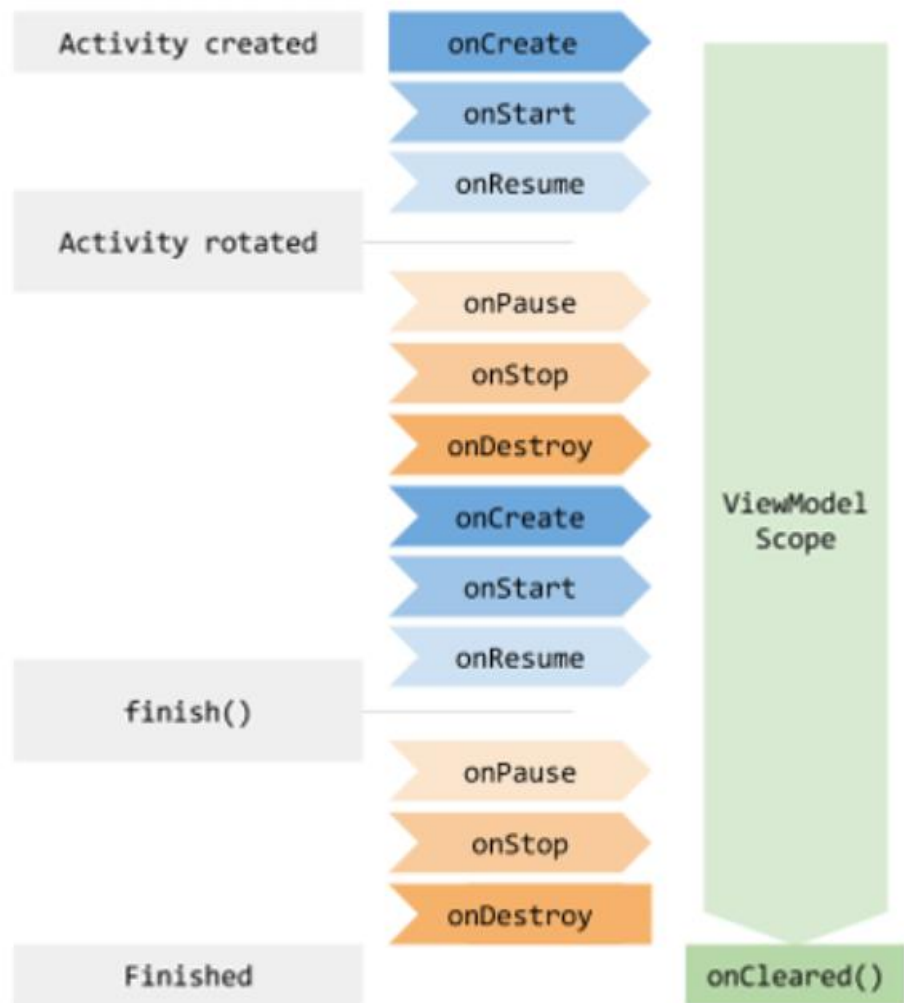
Dziękujemy
za
uwagę



*



Jak to wygląda?



Android

Systemy operacyjne na
platformach mobilnych

O czym
będziemy
mówić

Co to
jest?

Jak to
wygląda?

Czy to jest
praktyczne?

Zastosowanie

Dziękujemy
za
uwagę

Czy to jest
praktyczne?

*

*

Klasa aktywności

```
public class UserActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.user_activity_layout);
        final UserModel viewModel = ViewModelProviders.of(this).get(UserModel.class);
        viewModel.userLiveData.observe(this, new Observer() {
            @Override
            public void onChanged(@Nullable User data) {
                // update ui.
            }
        });
        findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                viewModel.doAction();
            }
        });
    }
}
```

Klasa modelu

```
public class UserModel extends ViewModel {  
    public final LiveData<User> userLiveData = new LiveData<>();  
  
    public UserModel() {  
        // trigger user load.  
    }  
  
    void doAction() {  
        // depending on the action, do necessary business logic calls and update the  
        // userLiveData.  
    }  
}
```


Android

Systemy operacyjne na
platformach mobilnych

O czym
będziemy
mówić

Co to
jest?

Jak to
wygląda?

Czy to jest
praktyczne?

Zastosowanie

Dziękujemy
za
uwagę

Zastosowanie

The diagram features a large, light blue circle with a white border, centered on a dark blue background. Inside this circle, the word 'Zastosowanie' is written in a large, black, sans-serif font. Below the word, there are two smaller, solid blue circles arranged side-by-side. The left circle contains the text 'Najprostszy sposób?' and the right circle contains the text 'Najlepszy sposób?'. Faint, light pink lines and dots are visible within the large circle, suggesting a network or flowchart structure. The background is decorated with a pattern of dark blue lines and dots, some of which are connected by lines, creating a technical or digital aesthetic.

Najprostszy
sposób?

Najlepszy
sposób?

Najprostszym sposobem jest co innego

Dla prostych danych najlepiej jest użyć zestawu funkcji onSaveInstanceState(), oraz onRestoreInstanceState().



```
19 @Override
20 public void onSaveInstanceState(Bundle savedInstanceState) {
21     TextView resultBox = findViewById(R.id.resultBox);
22     savedInstanceState.putString("savedState", resultBox.getText());
23     savedInstanceState.putDouble("leftNum", leftNumber);
24     savedInstanceState.putString("operation", operationType);
25     savedInstanceState.putDouble("rightNum", rightNumber);
26     savedInstanceState.putBoolean("state", isDone);
27     super.onSaveInstanceState(savedInstanceState);
28 }
29
30
31 @Override
32 public void onRestoreInstanceState(Bundle savedInstanceState) {
33     super.onRestoreInstanceState(savedInstanceState);
34     TextView resultBox = findViewById(R.id.resultBox);
35     leftNumber = savedInstanceState.getDouble(key: "leftNum");
36     savedState = savedInstanceState.getString(key: "savedState");
37     operationType = savedInstanceState.getString(key: "operation");
38     rightNumber = savedInstanceState.getDouble(key: "rightNum");
39     isDone = savedInstanceState.getBoolean(key: "state");
40     if(operationType.equals("none"))
41         changeButtonsState(true);
42     else
43         changeButtonsState(false);
44     resultBox.setText(savedState);
45 }
46
47 @Override
48 protected void onCreate(Bundle savedInstanceState) {
49     super.onCreate(savedInstanceState);
50     setContentView(R.layout.activity_simple_calc);
51 }
```


Najlepszym sposobem jest ViewModels

```
1  public class DetailActivityViewModel extends ViewModel {  
2      public Details mDetails;  
3  
4      public Details getDetails() {  
5          return mDetails;  
6      }  
7  
8      public void setDetails(Details details) {  
9          mDetails = details;  
10     }  
11 }
```

DetailActivityViewModel.java hosted with ❤️ by GitHub

[view raw](#)

Android

Systemy operacyjne na
platformach mobilnych

O czym
będziemy
mówić

Co to
jest?

Jak to
wygląda?

Czy to jest
praktyczne?

Zastosowanie

Dziękujemy
za
uwagę

Źródła

[https://medium.com/mindorks/
android-viewmodel-came-to-rescue-
transactiontoolargeexception-
ded4c5e2fc13](https://medium.com/mindorks/android-viewmodel-came-to-rescue-transactiontoolargeexception-ded4c5e2fc13)

[https://developer.android.com/topic/
libraries/architecture/
viewmodel.html](https://developer.android.com/topic/libraries/architecture/viewmodel.html)

[https://developer.android.com/
reference/android/arch/lifecycle/
ViewModel.html](https://developer.android.com/reference/android/arch/lifecycle/ViewModel.html)