WSTĘP DO ANDROIDA

Laboratorium 1

Systemy i aplikacje bez granic

II

- Uruchomić Android Studio

II

- Uruchomić Android Studio
- Stworzyć projekt typu Empty Activity

II

- Uruchomić Android Studio
- Stworzyć projekt typu Empty Activity
- Zapoznać się z projektantem widoku (activity_main.xml)

II

- Uruchomić Android Studio
- Stworzyć projekt typu Empty Activity
- Zapoznać się z projektantem widoku (activity_main.xml)
- i kodem klasy sterującej widokiem (MainActivity.kt)

III

- Stworzyć projekt typu Empty Activity

III

- Stworzyć projekt typu Empty Activity
- Usunąć z projektu TextView i wstawić w to miejsce Plain Text

III

- Stworzyć projekt typu Empty Activity
- Usunąć z projektu TextView i wstawić w to miejsce Plain Text
- Zmienić atrybut inputType na wartość ,,text”

III

- Stworzyć projekt typu Empty Activity
- Usunąć z projektu TextView i wstawić w to miejsce Plain Text
- Zmienić atrybut inputType na wartość ,,text”
- Usunąć ,,Name” z atrybutu ,,text”

III

- Dodać do klasy „import android.util.Log”

III

- Dodać do klasy „import android.util.Log”
- W klasie zdefiniować stałą val TAG = "StateChange"

III

- Dodać do klasy „import android.util.Log”
- W klasie zdefiniować stałą val TAG = "StateChange"
- Na koniec metody onCreate dodać instrukcję Log.i(TAG, „onCreate")

III

- Dodać do klasy „import android.util.Log”
- W klasie zdefiniować stałą val TAG = "StateChange"
- Na koniec metody onCreate dodać instrukcję Log.i(TAG, „onCreate")
- Uruchomić program

III

- Przeciążyć wszystkie pozostałe metody typu on… (onRestart, onStart, onResume, onPause, onStop, onDestroy, onConfigurationChanged, onRestoreInstanceState, on SaveInstanceState) dodając do logu odpowiedni komunikat

III

- Przeciążyć wszystkie pozostałe metody typu on… (onRestart, onStart, onResume, onPause, onStop, onDestroy, onConfigurationChanged, onRestoreInstanceState, on SaveInstanceState) dodając do logu odpowiedni komunikat

- Uruchamiać program testując czas życia aplikacji (uruchomienie, zawieszenie, przywrócenie, zabicie, zmiana konfiguracji)

IV

IV

- Stworzyć nowy projekt pt. Layout Example z szablonu „Add No Activity”

IV

- Stworzyć nowy projekt pt. Layout Example z szablonu „Add No Activity”
- Rozwinąć projekt do węzła app/java/*.layoutexample

IV

- Stworzyć nowy projekt pt. Layout Example z szablonu „Add No Activity”

- Rozwinąć projekt do węzła app/java/*.layoutexample

- Kliknąć prawym klawiszem i wybrać New/Activity/EmptyActivity

IV

- Stworzyć nowy projekt pt. Layout Example z szablonu „Add No Activity”
- Rozwinąć projekt do węzła app/java/*.layoutexample
- Kliknąć prawym klawiszem i wybrać New/Activity/EmptyActivity
- Nazwać ją „LayoutSample” i dodać do projektu

IV

IV

- Otworzyć app/manifests/AndroidManifest.xml

IV

- Otworzyć app/manifests/AndroidManifest.xml
- Dopisać do Activity

IV

- Otworzyć app/manifests/AndroidManifest.xml
- Dopisać do Activity

```
<activity android:name=".LayoutSample">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category
            android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

IV

IV

- Ściągnąć obrazek
  su22.png (https://tinyurl.com/ycrdmfka)

IV

- Ściągnąć obrazek su22.png (https://tinyurl.com/ycrdmfka)
- Dodać go do folderu app/res/drawable

IV

- Ściągnąć obrazek su22.png (https://tinyurl.com/ycrdmfka)
- Dodać go do folderu app/res/drawable
- Dodać na środku widoku ImageView i wskazać dodany wcześniej obrazek

IV

- Ściągnąć obrazek su22.png (https://tinyurl.com/ycrdmfka)
- Dodać go do folderu app/res/drawable
- Dodać na środku widoku ImageView i wskazać dodany wcześniej obrazek
- Nad ImageView dodać TextView. Ustawić tekst na wycentrowany, 24p, tekst=„SU-22”

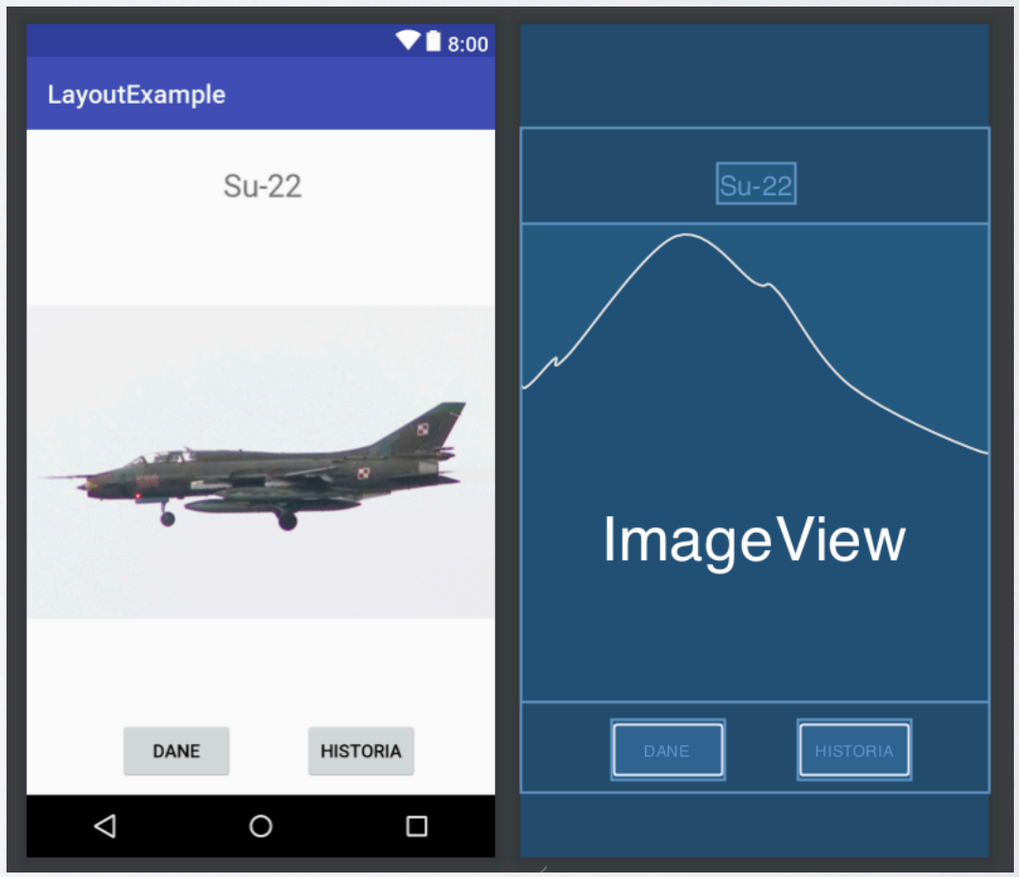
IV

IV

- Poniżej ImageView dodać obok siebie 2 przyciski Button:
  - Dane
  - Historia
- Uruchomić

IV

- Poniżej ImageView dodać obok siebie 2 przyciski Button:
  - Dane
  - Historia
- Uruchomić



IV

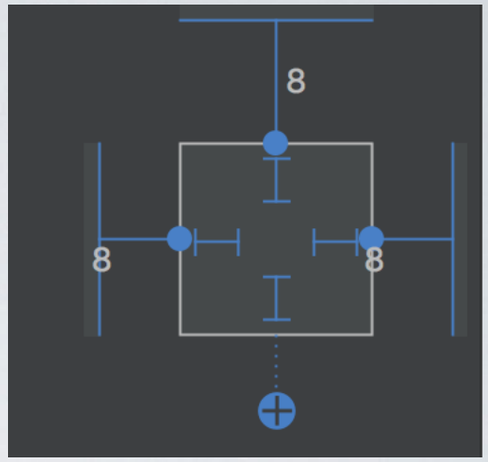
IV

- Dodajemy ograniczenia dla TextView

IV

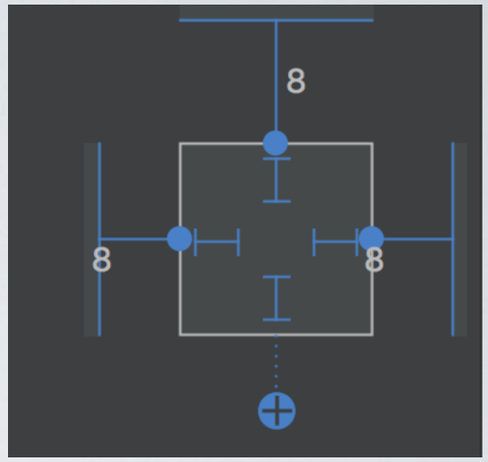
- Dodajemy ograniczenia dla TextView

IV



- Dodajemy ograniczenia dla TextView

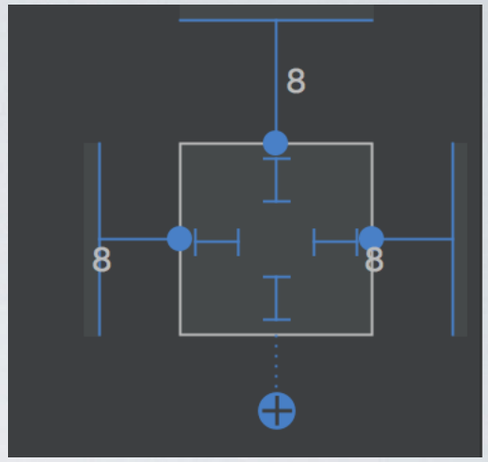
IV



- Dodajemy ograniczenia dla TextView

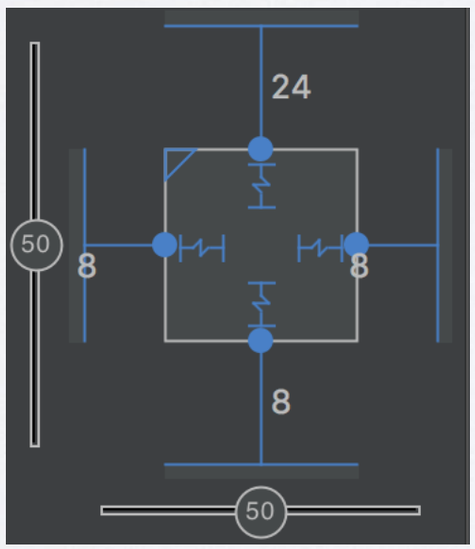
- Dla ImageView

IV

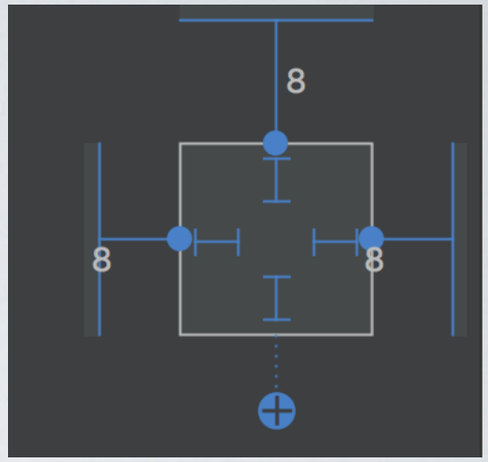


- Dodajemy ograniczenia dla TextView

- Dla ImageView

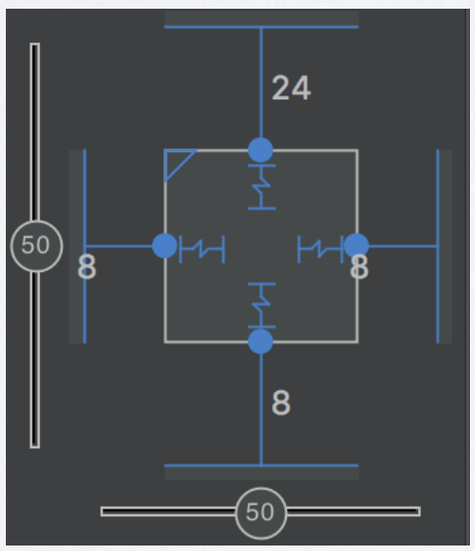


IV



- Dodajemy ograniczenia dla TextView

- Dla ImageView



IV

IV

- I dla przycisków

IV

- I dla przycisków
  - Najpierw zaznaczamy je razem (Shift+Click)

IV

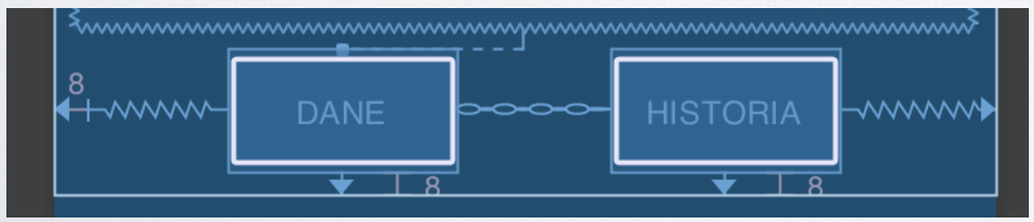
- I dla przycisków
  - Najpierw zaznaczamy je razem (Shift+Click)
  - Potem robimy RightClick i wybieramy Chain\Create Horizontal Chain

IV

- I dla przycisków
  - Najpierw zaznaczamy je razem (Shift+Click)
  - Potem robimy RightClick i wybieramy Chain\Create Horizontal Chain
  - Na koniec odstęp od dołu ekranu

IV

- I dla przycisków
  - Najpierw zaznaczamy je razem (Shift+Click)
  - Potem robimy RightClick i wybieramy Chain\Create Horizontal Chain
  - Na koniec odstęp od dołu ekranu



V

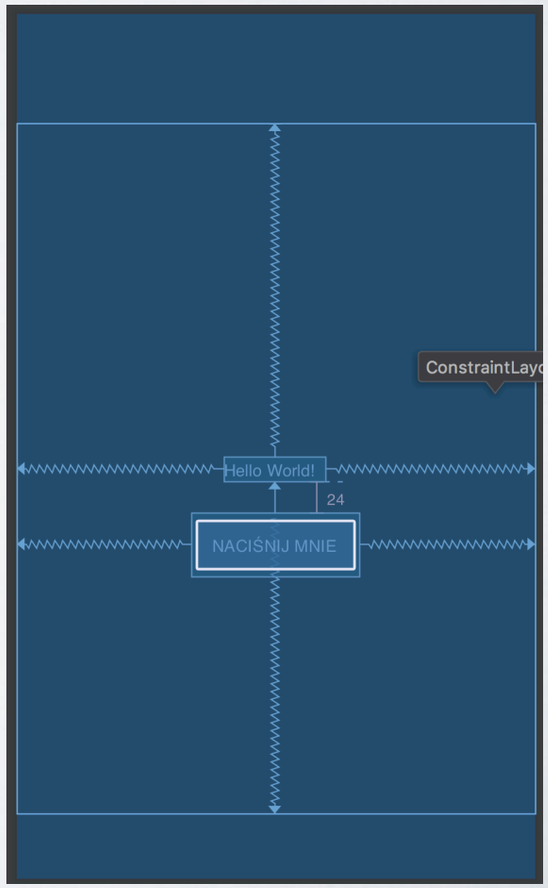
- Utworzyć nowy projekt typu Empty Activity

V

- Utworzyć nowy projekt typu Empty Activity
- Pod napisem Hello World dodać przycisk Button

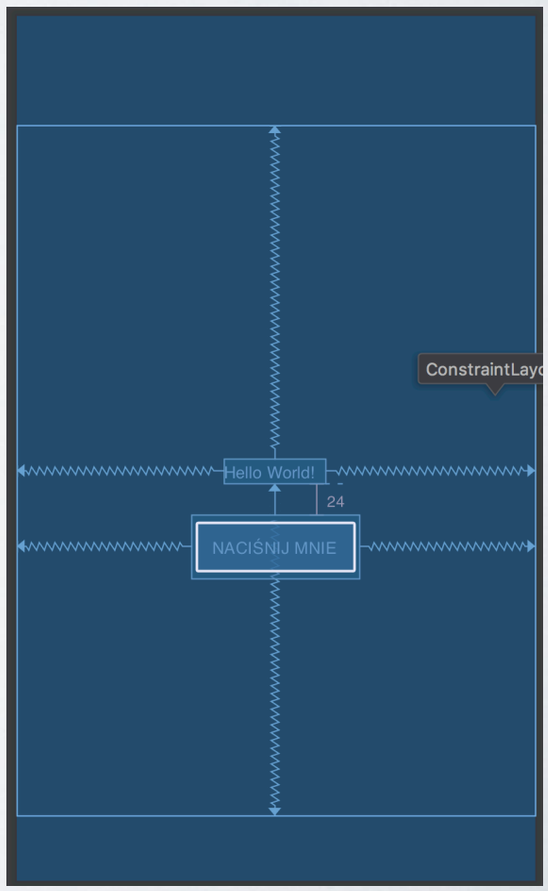
V

- Utworzyć nowy projekt typu Empty Activity
- Pod napisem Hello World dodać przycisk Button



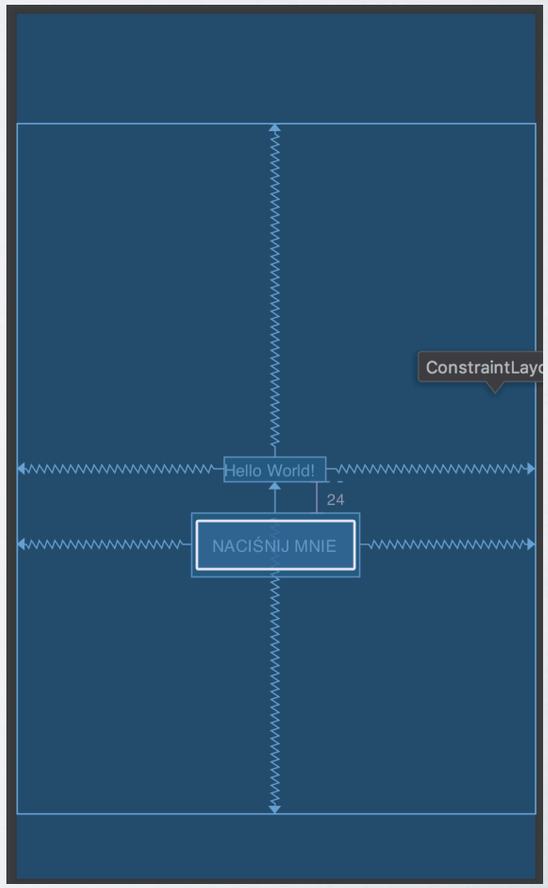
V

- Utworzyć nowy projekt typu Empty Activity
- Pod napisem Hello World dodać przycisk Button
- Zmienić id TextView na statusText



V

- Utworzyć nowy projekt typu Empty Activity
- Pod napisem Hello World dodać przycisk Button
- Zmienić id TextView na statusText
- i przycisku na pressMeButton



V

- W klasie widoku musimy dodać import na projekt

V

- W klasie widoku musimy dodać import na projekt

```
import kotlinx.android.synthetic.main.activity_main.*
```

V

- W klasie widoku musimy dodać import na projekt

```
import kotlinx.android.synthetic.main.activity_main.*
```

- oraz obsługę zdarzenia

V

- W klasie widoku musimy dodać import na projekt

```
import kotlinx.android.synthetic.main.activity_main.*
```

- oraz obsługę zdarzenia

```
pressMeButton.setOnClickListener {
    statusText.text = "Naciśnieto"
}
```

V

- W klasie widoku musimy dodać import na projekt

```
import kotlinx.android.synthetic.main.activity_main.*
```

- oraz obsługę zdarzenia

```
pressMeButton.setOnClickListener {
    statusText.text = "Naciśnieto"
}
```

- Uruchomić

V

- W klasie widoku musimy dodać import na projekt

```
import kotlinx.android.synthetic.main.activity_main.*
```

- oraz obsługę zdarzenia

```
pressMeButton.setOnClickListener {
    statusText.text = "Naciśnieto"
}
```

- Uruchomić

- Możemy jeszcze dodać

V

- W klasie widoku musimy dodać import na projekt

```
import kotlinx.android.synthetic.main.activity_main.*
```

- oraz obsługę zdarzenia

```
pressMeButton.setOnClickListener {
    statusText.text = "Naciśnięto"
}
```

- Uruchomić

- Możemy jeszcze dodać

```
pressMeButton.setOnLongClickListener {
    statusText.text = "Długie wciśnięcie"
    true
}
```