



Aplikacje mobilne

Proste aplikacje, aktywności.

dr Tomasz Jach
Instytut Informatyki,
Uniwersytet Śląski



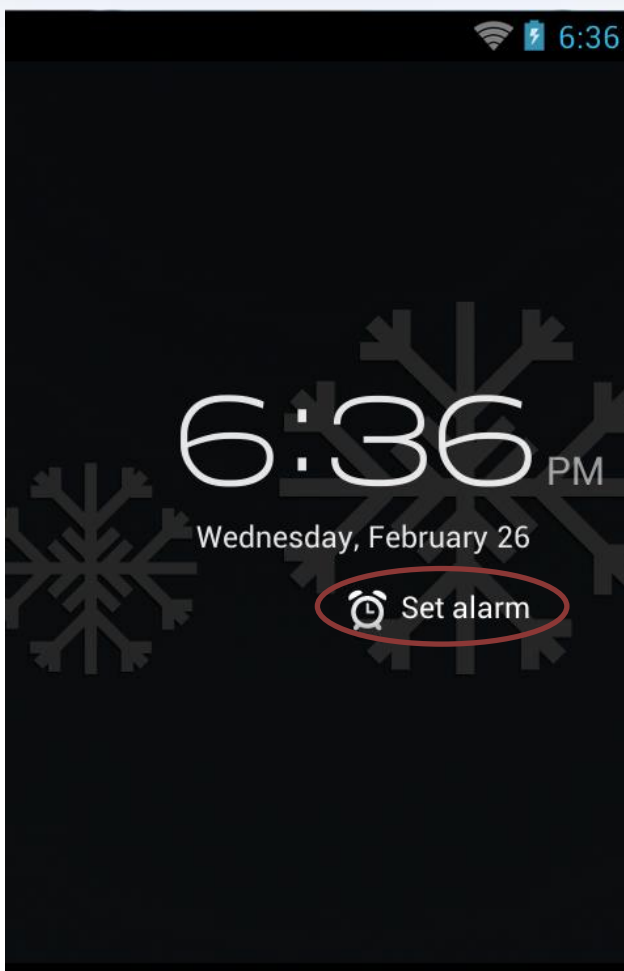
Activity (aktywność)

- Zwykle aplikacja składa się z jednej lub więcej aktywności.
- Tylko jedna aktywność (zwana główną) jest wybrana do wyświetlania przy pierwszym uruchomieniu aplikacji.
- Aktywność może przekazać sterowanie (i dane) do innej aktywności wykorzystując protokół komunikacyjny zwany intencją (intent).
- Aktywność zazwyczaj utożsamiana jest z pojedynczym ekranem GUI.

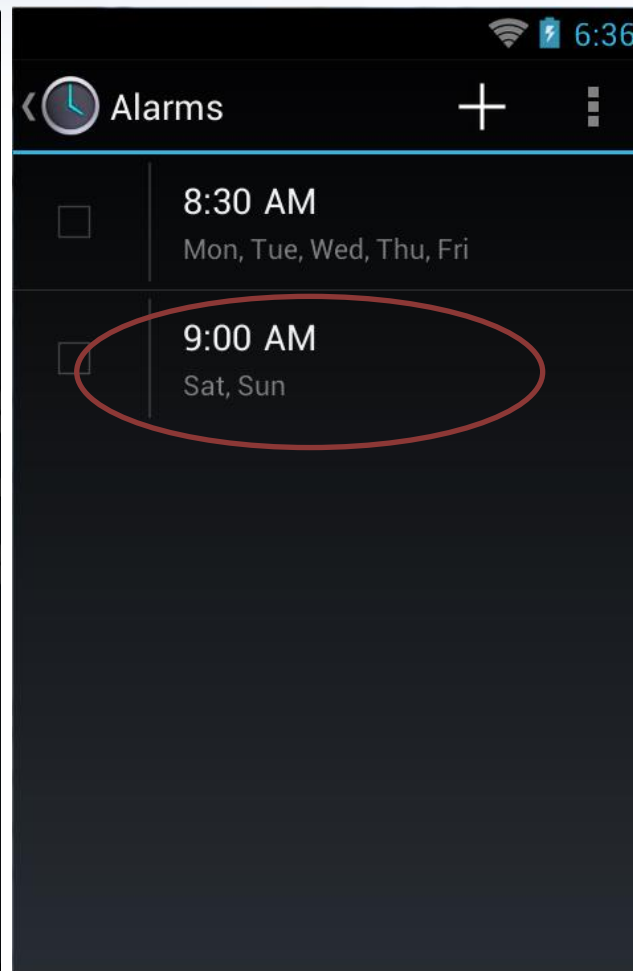
Przykład trzech aktywności



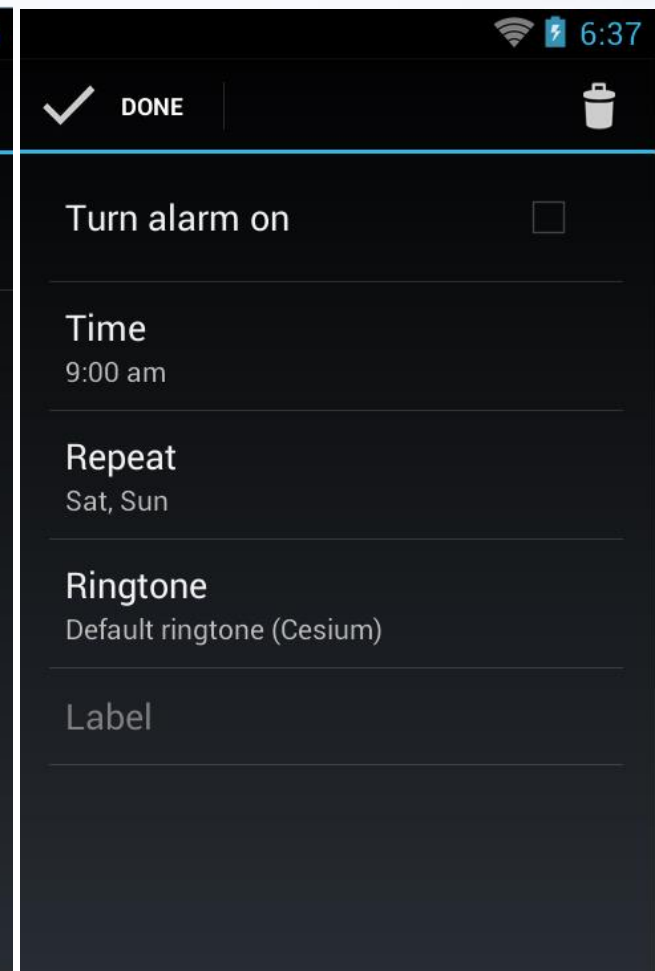
Aktywność 1



Aktywność 2



Aktywność 3



Implementacja callbacks



```
public class PrzykladAktywnosci extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Aktywność jest tworzona
    }
    @Override
    protected void onStart() {
        super.onStart();
        // Aktywność będzie widoczna
    }
    @Override
    protected void onResume() {
        super.onResume();
        // Aktywność jest widoczna (na pierwszym planie)
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Inna aktywność posiada focus
    }
    @Override
    protected void onStop() {
        super.onStop();
        // Aktywność nie jest już widoczna
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // Aktywność zostanie usunięta
    }
}
```

Zakończenie cyklu życia



- Metody **onPause()**, **onStop()**, **onDestroy()** mają status *killable*, czyli po zakończeniu dowolnej z nich, pozostałe nie muszą zostać wywołane (jeśli system wymusi zakończenie aplikacji).
- **onPause()** to jedyna metoda, która na pewno będzie wywołana przed zakończeniem aplikacji.
- Metoda **onPause()** powinna być wykorzystywana do zapisania stanu aplikacji.

Badanie cyklu życia aktywności - przykład



```
package pl.tomaszx.helloworld;

import android.os.Bundle;

public class MainActivity extends Activity {
    private Context context;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnExit = (Button) findViewById(R.id.button1);

        btnExit.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        context = getApplicationContext();
        Toast.makeText(context, "onCreate", Toast.LENGTH_SHORT).show();
    }
}
```

Badanie cyklu życia aktywności – przykład c.d.



```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    Toast.makeText(context, "onDestroy", Toast.LENGTH_SHORT).show();  
}
```

```
@Override  
protected void onPause() {  
    super.onPause();  
    Toast.makeText(context, "onPause", Toast.LENGTH_SHORT).show();  
}
```

```
@Override  
protected void onRestart() {  
    super.onRestart();  
    Toast.makeText(context, "onRestart", Toast.LENGTH_SHORT).show();  
}
```

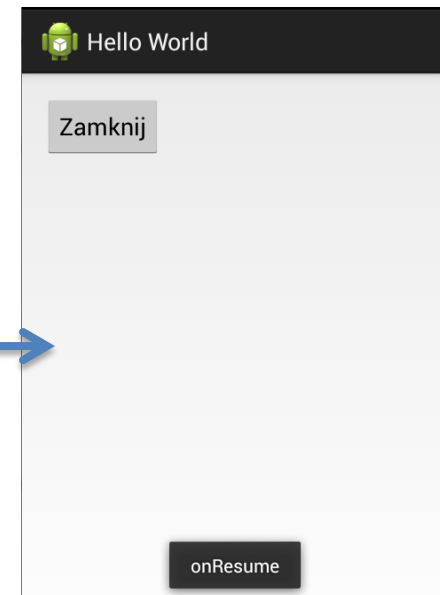
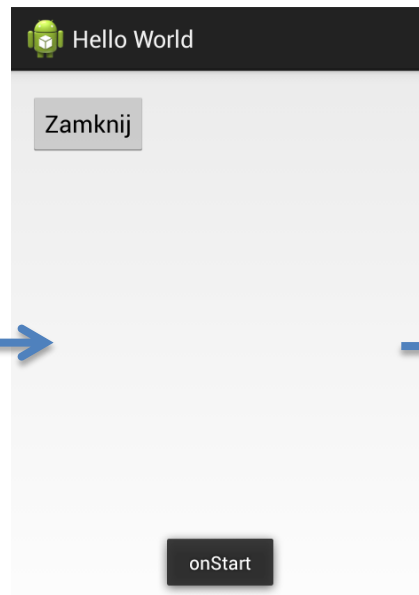
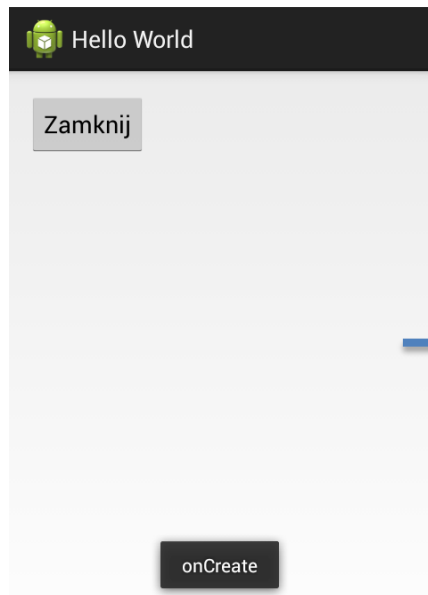
```
@Override  
protected void onResume() {  
    super.onResume();  
    Toast.makeText(context, "onResume", Toast.LENGTH_SHORT).show();  
}
```

Badanie cyklu życia aktywności – przykład c.d.



```
@Override  
protected void onStart() {  
    super.onStart();  
    Toast.makeText(context, "onStart", Toast.LENGTH_SHORT).show();  
}
```

```
@Override  
protected void onStop() {  
    super.onStop();  
    Toast.makeText(context, "onStop", Toast.LENGTH_SHORT).show();  
}
```





Zadanie 0

- Proszę zainstalować emulator Genymotion
<http://www.genymotion.com>
 - W tym celu należy się zarejestrować.
 - Dane do rejestracji będą później potrzebne!
 - Proszę pamiętać o wtyczce do Eclipse



Zadanie 1

- Napisz aplikację, która wyświetla komunikat
- w zależności od tego w jakim stanie znajduje
- się aktualnie aktywność.
- Komunikaty to:
 - onCreate
 - onStart
 - onRestart
 - onResume
 - onPause
 - onStop
 - onDestroy



Zadanie 2

- Na podstawie wiadomości zawartych w <http://andrzejklusiewicz-android.blogspot.com/2014/02/wywoywanie-aktywnosci-ekranow-elementy.html>

stworzyć prostą aplikację, której zadaniem będzie wyznaczanie pól podstawowych figur geometrycznych.



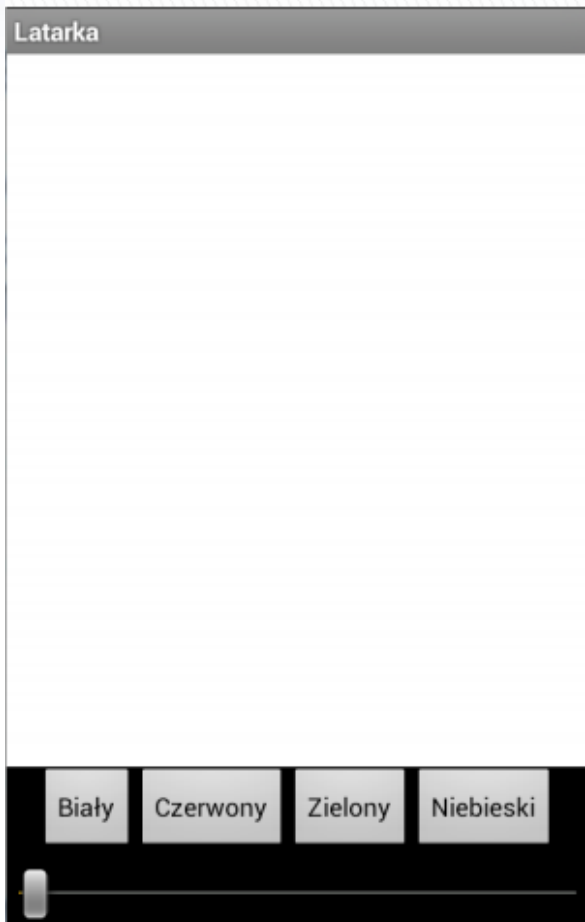
Zadanie 3

- Napisz następujące aplikacje, korzystając z podstawowych elementów GUI (Button, TextView, RadioButton):
 - Do symulowania latarki (poprzez zmianę koloru tła całej aktywności na jeden z 4 kolorów).
 - Do konwersji temperatury między Celsjuszem a Kelwinem i Fahrenheitem.
 - Do obliczania średniego spalania paliwa na 100 km wraz z jego kosztem.



Zadanie 3a

Latarka – pomocne wiadomości



```
SeekBar sBarOpacity = (SeekBar) findViewById(R.id.sbrPrzezroczytosc);
sBarOpacity.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {

    public void onStopTrackingTouch(SeekBar seekBar) {

    }

    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        alpha = 255 - progress;
        int myBackColor = android.graphics.Color.argb(alpha, red, green, blue);
        tvLight.setBackgroundColor(myBackColor);
    }
});
```

Pamiętać o dodaniu implements OnClickListener

```
public void onClick(View v) {
    if (v.getId() == btnWhite.getId())
    {
        //KOD OBSŁUGI
    }
}
```



Zadanie 4

- Napisz grę „Zgadnij moją liczbę”. Program losuje liczbę z zakresu $[-2000;2000]$ a zadaniem użytkownika jest ją zgadnąć (na podstawie odpowiedzi „za dużo”, „za mało”). Program wyświetla ponadto liczbę prób.
- Zadbaj o wygląd, layout, efekty graficzne po wygranej (nowy ekran aktywności)
- Zabezpiecz przed wpisywaniem wartości spoza zakresu (komunikaty wyświetlane `Toast()`)

Zadanie 5

- Napisz aplikację symulującą zachowanie świateł drogowych.
- Aplikacja powinna wykorzystywać komponenty ImageView i Button.
- Po kliknięciu na przycisk zapala się odpowiednie światło.
- Zadbaj o kontrolę błędów – nie można zapalić trzech świateł jednocześnie.
- Zmiany obrazka dla ImageView dokonuje się funkcją setImageResource. Obrazki narysuj w programie graficznym.

