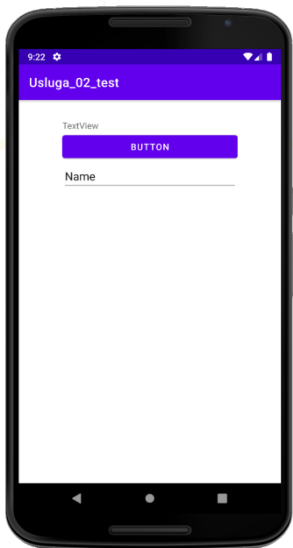


Wątki i Usługi

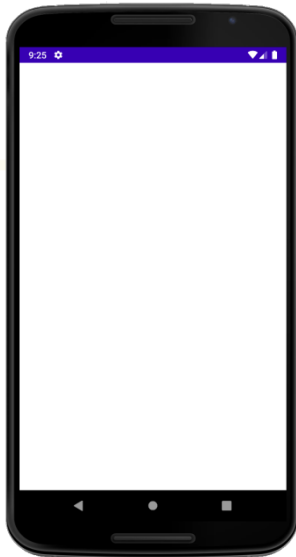
PUM

Chcemy zrobić zegar w aplikacji

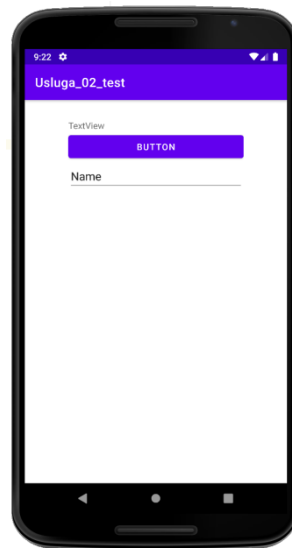
```
do {  
    SystemClock.sleep(1000);  
    textViewTime.setText(new SimpleDateFormat("HH:mm:ss", Locale.getDefault()).format(new Date()));  
} while (true);
```



Kod w metodzie
onCreate(...),
efekt →



Kod w metodzie
onButtonClick(...),
efekt →



Rozwiązanie 1 – nowy wątek Thread

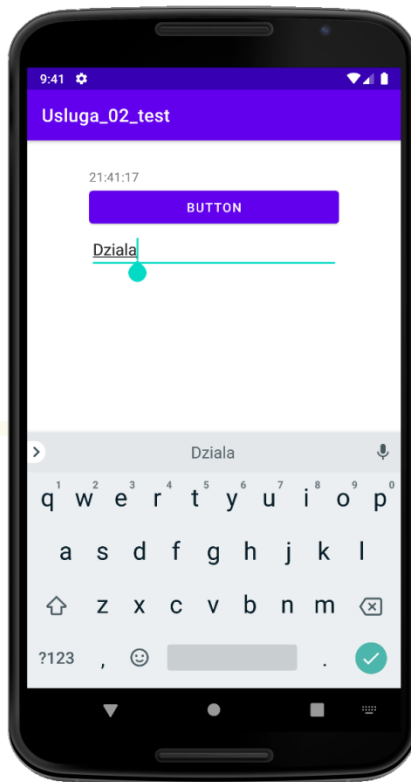
```
new Thread(new Runnable() {  
    @Override  
    public void run() {  
        do {  
            textViewTime.setText(new SimpleDateFormat("HH:mm:ss",  
                Locale.getDefault()).format(new Date()));  
            SystemClock.sleep(1000);  
        } while(true);  
    }  
}).start();
```

Kod w metodzie
onButtonClick(...),
efekt →

Równoważny kod wykorzystujący notację lambda

```
new Thread(() -> {  
    do {  
        textViewTime.setText(new SimpleDateFormat("HH:mm:ss",  
            Locale.getDefault()).format(new Date()));  
        SystemClock.sleep(1000);  
    } while(true);  
}).start();
```

Kod w metodzie
onCreate(...),
efekt →



Kod nowego wątku drukującego wiadomości na ekranie i w dzienniku

```
final TextView textView = findViewById(R.id.textView3);

new Thread(new Runnable(){
    public void run(){
        //textView = findViewById(R.id.textView3);
        textView.setText("Start wątku\n" );
        for (int i=1; i<=20; i++) {
            try {
                Thread.sleep(1500);
                textView.setText("Wątek: i = " + i + "\n" );
                Log.v("Program->Wątek", "Run" + i);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        textView.setText("Koniec wątku\n" );
    }
}).start();
```

Efekt

```
Log.v("Program->Wątek", "Run" + i);
```

Kod w metodzie `onButtonClick(...)`, Przycisk kliknięty wielokrotnie

Logcat ->

```
2023-01-24 02:06:27.707 8670-8704/com.example.usługa_01 V/Program->Wątek: Run6
2023-01-24 02:06:27.759 8670-8701/com.example.usługa_01 V/Program->Wątek: Run17
2023-01-24 02:06:27.860 8670-8703/com.example.usługa_01 V/Program->Wątek: Run11
2023-01-24 02:06:29.210 8670-8704/com.example.usługa_01 V/Program->Wątek: Run7
2023-01-24 02:06:29.265 8670-8701/com.example.usługa_01 V/Program->Wątek: Run18
2023-01-24 02:06:29.362 8670-8703/com.example.usługa_01 V/Program->Wątek: Run12
2023-01-24 02:06:30.718 8670-8704/com.example.usługa_01 V/Program->Wątek: Run8
2023-01-24 02:06:30.773 8670-8701/com.example.usługa_01 V/Program->Wątek: Run19
2023-01-24 02:06:30.869 8670-8703/com.example.usługa_01 V/Program->Wątek: Run13
2023-01-24 02:06:32.227 8670-8704/com.example.usługa_01 V/Program->Wątek: Run9
2023-01-24 02:06:32.277 8670-8701/com.example.usługa_01 V/Program->Wątek: Run20
2023-01-24 02:06:32.373 8670-8703/com.example.usługa_01 V/Program->Wątek: Run14
2023-01-24 02:06:33.731 8670-8704/com.example.usługa_01 V/Program->Wątek: Run10
2023-01-24 02:06:33.877 8670-8703/com.example.usługa_01 V/Program->Wątek: Run15
2023-01-24 02:06:35.235 8670-8704/com.example.usługa_01 V/Program->Wątek: Run11
2023-01-24 02:06:35.380 8670-8703/com.example.usługa_01 V/Program->Wątek: Run16
2023-01-24 02:06:36.737 8670-8704/com.example.usługa_01 V/Program->Wątek: Run12
```

Rozwiązanie 2 – usługa Service

Usługi uruchomione (*started services*) mogą działać w tle dowolnie długi czas, nawet po tym, jak aktywność, która ją uruchomiła zostanie usunięta. Po zakończeniu zadania usługa jest zatrzymywana.

Usługi powiązane (*bounded services*) są skojarzone z konkretnym komponentem, takim jak aktywność. Aktywność może prowadzić interakcję z usługą i przysyłać do niej żądania i pobierać wyniki. Usługi tego typu działają tak długo, jak długo działa komponent, z którym są powiązane.

IntentService

- Nowa aplikacja
- Nowa usługa
 - File/New/Service/Service(IntentService)

Gotowe dwie metody do wykorzystania

```
/**
 * Handle action Foo in the provided background thread with the provided
 * parameters.
 */
private void handleActionFoo(String param1, String param2) {
    // TODO: Handle action Foo
    throw new UnsupportedOperationException("Not yet implemented");
}

/**
 * Handle action Baz in the provided background thread with the provided
 * parameters.
 */
private void handleActionBaz(String param1, String param2) {
    // TODO: Handle action Baz
    throw new UnsupportedOperationException("Not yet implemented");
}
```

Korzystanie z metod usługi

Wywołanie metody *startActionBaz* w aktywności głównej

```
public void onClickStart(View v) {  
    MyIntentService.startActionBaz(getApplicationContext(),"Parametr x 1","Parametr y 2" );  
}
```

Obsługa akcji *ActionBaz* w kodzie usługi

```
private void handleActionBaz(String param1, String param2) {  
    for(int i = 10; i < 33; i++) {  
        SystemClock.sleep(1000);  
        Log.v("usluga_13", param1 + " nr " + i);  
        MainActivity.textViewSt.setText("Ala ma kota. " + param2 + " nr " + i);  
    }  
    // throw new UnsupportedOperationException("Not yet implemented");  
}
```

Dostęp do kontrolek w MainActivity przez zmienne statyczne: `static` TextView *textViewSt*;