
Słuchacz zdarzeń



Słuchacz zdarzeń

- ✓ Słuchacz zdarzeń to komponent, który odbiera zdarzenie (np. kliknięcie) i reaguje na nie.
- ✓ Słuchacze zdarzeń to klasy posiadające metody obsługi poszczególnych zdarzeń. Metody są wywoływane, gdy nastąpi odpowiadające im zdarzenie (np. rozpoczęcie przesuwania suwaka, zakończenie przesuwania suwaka itp.)
- ✓ Słuchacze są powiązane z określonymi kontrolkami. Np. domyślnym słuchaczem dla kontrolki Button jest `onClickListener`, a dla kontrolki SeekBar - `onSeekBarChangeListener`

Obsługa komponentu Button za pomocą słuchacza zdarzeń

`View.OnClickListener`. `listenr`

Istnieją trzy najczęściej stosowane metody obsługi kliknięcia na przycisk:

1. Dodanie do jego opisu w pliku XML parametru **onClick** informującego jaką metodę należy wywołać po kliknięciu (tak obsługiwaliśmy przyciski we wcześniejszych przykładach)
2. Utworzenie dla każdego przycisku jego własnego **słuchacza zdarzeń**,
3. Zarejestrowanie przycisku do „zbiórczego” **słuchacza zdarzeń** obsługującego kilka przycisków.

Aplikacje mobilne

Utworzenie dla przycisku jego własnego słuchacza zdarzeń.

```
10 public class MainActivity extends AppCompatActivity {
11
12     Button przycisk;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         przycisk = findViewById(R.id.button01);
20         przycisk.setOnClickListener(new View.OnClickListener() {
21             @Override
22             public void onClick(View view) {
23                 // akcja po kliknięciu na przycisk
24             }
25         });
26     }
27 }
```

Tworzymy pustą referencję do przycisku.

Uzyskujemy referencję przycisku.

Do przycisku dodajmy słuchacza – metoda `setOnClickListener()` której parametrem jest sam listener (klasa anonimowa – cała opisana w i od razu użyta jako parametr wywołania metody)

Ten typ słuchacza posiada tylko jedną metodą `onClick()`, która wywoływana jest w momencie kliknięcia na przycisk – tu wstawiamy kod

Aplikacje mobilne

Utworzenie dla przycisku jego własnego słuchacza zdarzeń.

PODPOWIEDŹ: Tworząc słuchacza należy użyć kreatora

```
przycisk.setOnClickListener(new );
```

Ctrl + Spacja

```
przycisk.setOnClickListener(new );
```

```
View.OnClickListener[...] (android.view.View)  
QuickContactBadge (android.widget)  
CharacterPickerDialog (android.text.method)  
ActionMenuItemView (androidx.appcompat.view...  
KeyboardView (android.inputmethodservice)  
boolean []  
byte []  
char []  
double []  
float []  
int []  
long []
```

Press Enter to insert, Tab to replace Next Tip

Aplikacje mobilne

Zarejestrowanie przycisku do „zbiornego” słuchacza zdarzeń obsługującego kilka przycisków.

Tworzymy obiekt klasy `View.OnClickListener`.
Listenerowi nadano nazwę „słuchacz”

Podobnie jak w poprzednim przykładzie posiada on metodę `onClick()`, którą oprogramujemy.

Referencja obiektu, na rzecz którego wywołany będzie listener przekazana jest w jego parametrze jako „view”. Z „view” wyciągamy id obiektu na rzecz którego wywołano listenera. Pozwal to rozpoznać, który przycisk został kliknięty.

Listener „słuchacz” ustawimy dla obu przycisków.

```
10 public class MainActivity extends AppCompatActivity {
11     Button przycisk1, przycisk2;
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         przycisk1 = findViewById(R.id.button01);
17         przycisk2 = findViewById(R.id.button02);
18         View.OnClickListener sluchacz = new View.OnClickListener() {
19             @Override
20             public void onClick(View view) {
21                 int id = view.getId();
22                 switch(id)
23                 {
24                     case R.id.button01: //alcja dla przycisku button1
25                         break;
26                     case R.id.button02: //alcja dla przycisku button2
27                         break;
28                 }
29             }
30         };
31         przycisk1.setOnClickListener(sluchacz);
32         przycisk2.setOnClickListener(sluchacz);
33     }
34 }
```



Aplikacje mobilne

Obsługa komponentu Button – słuchacz „długiego kliknięcia”

onLongClickListener. Listener

```
przycisk2.onLo  
    m onKeyDownPress(int keyCode, KeyEvent... boolean  
    m setOnLongClickListener(OnLongClickListener l) void  
przycisk  
    m hasOnLongClickListeners() boolean  
przycisk2.setOnLongClickListener(słuchacz),
```

Drugim, często używanym słuchaczem zdarzeń dla przycisku (i nie tylko) jest słuchacz **onLongClickListener()** który reaguje na „długie kliknięcie” czyli przytrzymanie elementu

Aplikacje mobilne

Obsługa komponentu Button – słuchacz „długiego kliknięcia”

onLongClickListener. Listenr

```
przycisk2.setOnLongClickListener(new View.OnLongClickListener() {  
    @Override  
    public boolean onLongClick(View view) {  
        // tu wpisz kod  
        return false;  
    }  
});
```

Słuchacz ten dostarcza metodę onLongClick, która wywoływana jest chwilę po przytrzymaniu widoku.