

Android Studio - Progress Bar

© Copyright by 3bird Projects 2024, <http://edukacja.3bird.pl>

Ogólne

Aplikacja prezentuje kilka typów tzw. „paska postępu”. Wersja rozbudowana zawiera dodatkowo symulację postępu pobierania pliku.

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępy. Kod należy przepisać ze zrozumieniem.

Kod - wersja podstawowa

```
package com.example.progressautorski
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.material3.CenterAlignedTopAppBar
import androidx.compose.material3.CircularProgressIndicator
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.LinearProgressIndicator
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TopAppBarDefaults
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.progressautorski.ui.theme.ProgressautorskiTheme
```

```

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView {
            ProgressautorskiTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    WszytkieElementy()
                }
            }
        }
    }
}

```

```
@OptIn(ExperimentalMaterial3Api::class)
```

```
@Composable
```

```
fun WszytkieElementy() {
```

```
    Column(
```

```
        verticalArrangement = Arrangement.spacedBy(12.dp),
```

```
        horizontalAlignment = Alignment.CenterHorizontally,
```

```
        modifier = Modifier.fillMaxWidth()
    ) {
```

```
        CenterAlignedTopAppBar(
```

```
            title = {
```

```
                Text(
```

```
                    "Progress Bars",
```

```
                    fontSize = 26.sp,
```

```
                    fontWeight = FontWeight.Bold
                )
            },
```

```
            colors = TopAppBarDefaults.centerAlignedTopAppBarColors(
```

```
                containerColor = Color.Blue.copy(alpha = 0.3f)
            )
        )
    }

```

```
    )
```

```
    Spacer(modifier = Modifier.height(5.dp))
```

```
    Text("Stacyjny liniowy:")
```

```
    LinearProgressIndicator()
```

```
    Spacer(modifier = Modifier.height(30.dp))
```

```
    Text("Stacyjny kołowy:")
```

```
    CircularProgressIndicator()
```

```
} // Koniec kolumny
```

```
}
```

Kod - wersja rozbudowana

```
package com.example.progressautorski
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.width
import androidx.compose.material3.Button
import androidx.compose.material3.CenterAlignedTopAppBar
import androidx.compose.material3.CircularProgressIndicator
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.LinearProgressIndicator
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TopAppBarDefaults
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.progressautorski.ui.theme.ProgressautorskiTheme
import java.lang.Thread.sleep
```

```

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView {
            ProgressatorskiTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    WszytkieElementy()
                }
            }
        }
    }
}

```

```
@OptIn(ExperimentalMaterial3Api::class)
```

```
@Composable
```

```
fun WszytkieElementy() {
```

```

    // Wartością zmiennej "aktualnyStanProgresu" jest procent zajętości paska postępu.
    // Na przykład, wartość 0.5f oznacza, że pasek postępu będzie zajęty w połowie (f = fill).
    // Początkowa wartość to 0%:

```

```

    var aktualnyStanProgresu by remember { mutableStateOf(0f) }
    var aktualnyStanProgresu2 by remember { mutableStateOf(0f) }
    var startLoading by remember { mutableStateOf(false) }
    var startLoading2 by remember { mutableStateOf(false) }

```

```
Column(
```

```

    verticalArrangement = Arrangement.spacedBy(12.dp),
    horizontalAlignment = Alignment.CenterHorizontally,
    modifier = Modifier.fillMaxWidth()

```

```
) {
```

```
CenterAlignedTopAppBar(
```

```

    title = {
        Text(
            "Progress Bars",
            fontSize = 26.sp,
            fontWeight = FontWeight.Bold
        )
    },

```

```

    colors = TopAppBarDefaults.centerAlignedTopAppBarColors(
        containerColor = Color.Blue.copy(alpha = 0.3f)
    )

```

```
)
```

```
Spacer(modifier = Modifier.height(5.dp))
```

```
Text("Statyczny liniowy:")
```

```
LinearProgressIndicator()
```

```
Spacer(modifier = Modifier.height(30.dp))
```

```

Text("Statyczny kołowy:")
CircularProgressIndicator()
Spacer(modifier = Modifier.height(50.dp))

// ===== PRZYCISK 1 =====
Text("Dynamiczny liniowy:")
// Uruchamiamy pasek postępu przypisując wartość true (nastąpi rekompozycja)
Button(
    onClick = { startLoading = true },
    // Przycisk jest aktywny tylko gdy startLoading = false
    enabled = !startLoading
) { Text("Start loading") }
// Koniec przycisku

// Ujawnienie paska progresu
if (startLoading) {
    LinearProgressIndicator(
        modifier = Modifier.fillMaxWidth(),
        color = Color.Green,
        progress = { aktualnyStanProgresu }
    )

    if (aktualnyStanProgresu < 1.0) {
        sleep(100)
        aktualnyStanProgresu += 0.01.toFloat()
    }
    else{
        startLoading = false
        aktualnyStanProgresu = 0f
    }
}

// ===== PRZYCISK 2 =====
Spacer(modifier = Modifier.height(30.dp))
Text("Dynamiczny kołowy:")
// Uruchamiamy pasek postępu przypisując wartość true (nastąpi rekompozycja)
Button(
    onClick = { startLoading2 = true },
    // Przycisk jest aktywny tylko gdy startLoading = false
    enabled = !startLoading2
) { Text("Start loading") }
// Koniec przycisku

// Ujawnienie paska progresu
if (startLoading2) {
    CircularProgressIndicator(
        modifier = Modifier.width(120.dp),
        color = Color.Green,

```

```

        progress = {aktualnyStanProgresu2}
    )

    if (aktualnyStanProgresu2 < 1.0) {
        sleep(100)
        AktualnyStanProgresu2 += 0.01.toFloat()
    }
    else{
        startLoading2 = false
        aktualnyStanProgresu2 = 0f
    }
}
} // Koniec kolumny
}

```

```

@Preview(showBackground = true)
@Composable
fun Podglad() {
    ProgressautorskiTheme {
        WszystkieElementy()
    }
}

```

Ostatnia aktualizacja: 14 lutego 2024.