Android Studio - Kotlin: AlertDialog

© Copyright by 3bird Projects 2026, http://edukacja.3bird.pl

Ogólne

Aplikacja generuje tzw. „*AlertDialog*", czyli przycisk zawierający dwie możliwe odpowiedzi: „*Tak*" oraz „*Nie*". W zależności od wyboru odpowiedzi, uruchamiane są różne instrukcje.

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii, twarde odstępy, odmienne cudzysłowy, odmienne apostrofy, odmienne myślniki. To godziny dodatkowej pracy na wykrywanie błędów i ich poprawianie. Kod należy przepisać ze zrozumieniem.

Kod

```
// W MainActivity:
Scaffold(
        topBar = { GorneMenu() },
        bottomBar = { DolneMenu() },
        modifier = Modifier.fillMaxSize()
) { innerPadding ->
        Glowna(innerPadding)
}


// Zewnętrzne funkcje:
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun GorneMenu() {
  TopAppBar(
    // Odstęp 10dp od górnej krawędzi paska nagłówka ekranu:
    modifier = Modifier.statusBarsPadding().padding(top = 10.dp),
    title = {
      // Aby tytuł był wyrównany do środka, należy najpierw rozciągnąć
      // pole tekstowe za pomocą "fillMaxWidth()":
      Text("Twoje wyzwania", modifier = Modifier.fillMaxWidth(),
        textAlign = TextAlign.Center, color = Color.Red)
    },
    // Ustawienie koloru tła nagłówka:
```

```
        colors = TopAppBarDefaults.topAppBarColors(containerColor = Color.DarkGray)
    )
}

@Composable
fun DolneMenu() {
    BottomAppBar { Text("(c) Copyright by Imię Nazwisko.", fontSize = 12.sp) }
}

@Composable
fun MojAlertDialog(
    // Funkcja posiada 4 parametry do pobrania argumentów:
    onDismissRequest: () -> Unit,
    onConfirmation: () -> Unit,
    dialogTitle: String,
    dialogText: String
) {
    // Predefiniowany element [AlertDialog]:
    AlertDialog(
        icon = {
            Icon(painter = painterResource(id = R.drawable.info_24px),
                contentDescription = "Example Icon")
        },
        title = { Text(text = dialogTitle) },
        text = { Text(text = dialogText) },
        onDismissRequest = { onDismissRequest() },
        confirmButton = { TextButton(
            onClick = { onConfirmation() }
            ) { Text("Potwierdzam") }
        },
        dismissButton = { TextButton(
            onClick = { onDismissRequest() }
            ) { Text("Odrzucam") }
        }
```

```
    )
}

@Composable
fun Glowna(innerPaddingValues: PaddingValues) {
    // Rozciąga kolumnę na cały dostępny ekran:
    Column(modifier = Modifier.fillMaxSize().padding(innerPaddingValues),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        // Domyślnie Dialog ma wartość false, czyli nie jest widoczny:
        val openAlertDialog = remember { mutableStateOf(false) }
        val contextForToast = LocalContext.current.applicationContext
        Text(text = "Hello Brajan!")
        Spacer(modifier = Modifier.height(30.dp))
        OutlinedButton(
            // Po kliknięciu w przycisk, wartość Dialogu zmienia się na przeciwną:
            onClick = { openAlertDialog.value = !openAlertDialog.value },
            colors = ButtonDefaults.buttonColors(
                // Predefiniowany jasny niebieski/cyjan:
                containerColor = Color.Cyan
                // Lub własny odcień błękitu:
                // containerColor = Color(0xFF87CEEB)
            )
        ) { Text("Dzisiejsze wyzwanie...") }
        Spacer(modifier = Modifier.height(30.dp))
        // Zamiast poniższego "when" można by zastosować "if (openAlertDialog.value) {..."
        when {
            // Gdy wartość Dialogu jest prawdziwa, wtedy otwórz go:
            openAlertDialog.value -> {
                MojAlertDialog(
                    // Przekazujemy funkcji MojAlertDialog cztery argumenty:
                    onDismissRequest = {
                        // Wartość "false" zamyka Dialog:
```

```
                openAlertDialog.value = false
                Toast.makeText(contextForToast, "Oj, niełądnie! Jestem tobą rozczarowany, mój
                    młody Brajanie.", Toast.LENGTH_LONG).show()
            },
            onConfirmation = {
                // Wartość "false" zamyka Dialog:
                openAlertDialog.value = false
                Toast.makeText(contextForToast, "No, bardzo dobry wybór!",
                    Toast.LENGTH_LONG).show()
            },
            dialogTitle = "Twoje zadanie...",
            dialogText = "Dzisiaj zorganizujesz sobie szmatkę / ręcznik papierowy i wyczyścisz
                monitor oraz klawiaturę. Powodzenia!"
        )
    }
  }
 }
}
```

Ocenianie:

- **dobry** - działający kod;

- **bardzo dobry** - dodatkowo własna ikona aplikacji;

- **celujący** - prezentacja aplikacji na fizycznym smartfonie oraz wykonanie zadania zaproponowanego przez aplikację.

Ostatnia aktualizacja: 2 listopada 2026.