

Dialogi

PUM

Okna dialogowe

- Klasyczne okno dialogowe służy do prezentowania użytkownikowi informacji i pozyskiwania od niego odpowiedzi.
- Klasy dialogów
 - ProgressDialog - okno postępu
 - AlertDialog - okno informacyjne
 - Dialog - dialog własny
- Funkcję podobną do dialogu może pełnić nowa intencja - aktywność

ProgressDialog

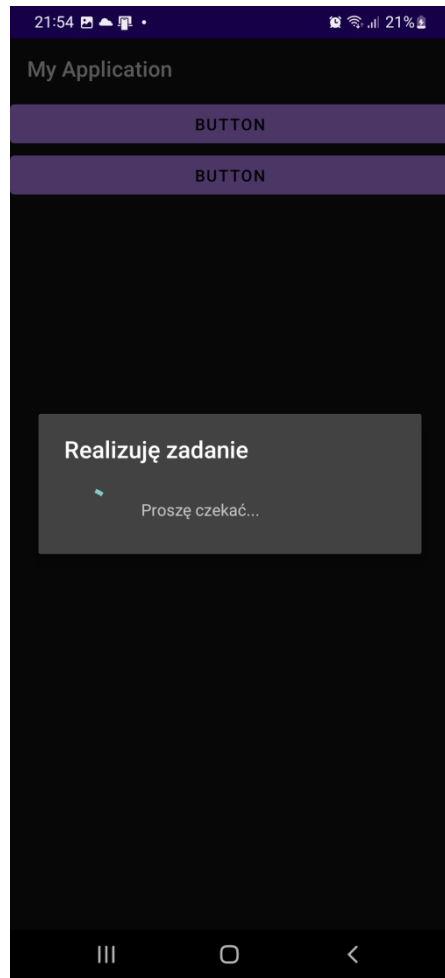
- Wyświetla małe okno pływające, znajdujące się nad aktualnym widokiem.
- Prezentuje użytkownikowi komunikat o postępie wraz z opcjonalnymi przyciskami.
- Dialog jest zamykany programowo po upływie czasu wykonywanego zadania lub poprzez kliknięcie danego przycisku.

Okno dialogowe postępu

```
public class MainActivity extends AppCompatActivity {

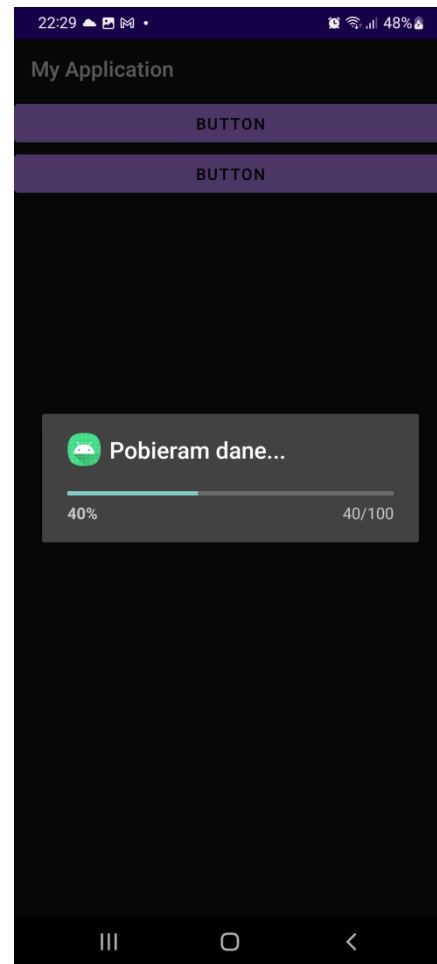
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ((Button)findViewById(R.id.button)).setOnClickListener(view -> OknoDialogowe(view));
    }

    public void OknoDialogowe(View view) {
        final ProgressDialog dialog = ProgressDialog.show(this,
            "Realizuję zadanie", "Proszę czekać...");
        new Thread(new Runnable(){
            public void run(){
                try {
                    Thread.sleep(3000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                dialog.dismiss();
            }
        }).start();
    }
}
```



Okno dialogowe postępu 2

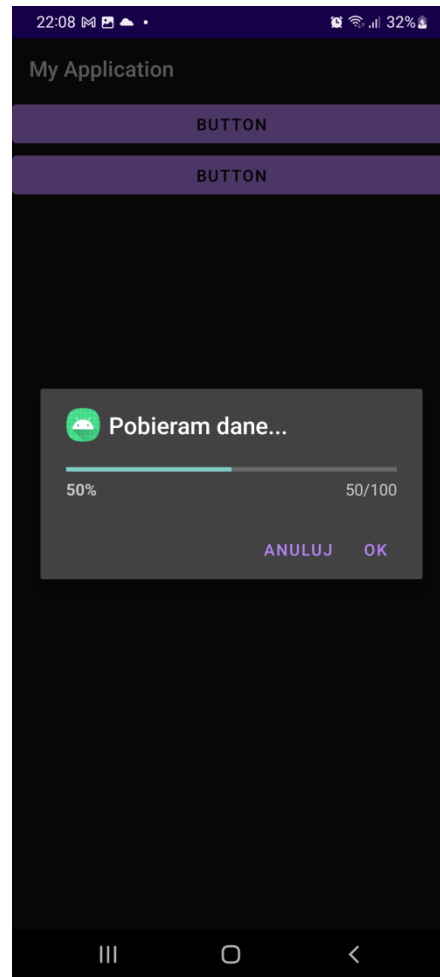
```
public void OknoPostepu(View v) {  
    final ProgressDialog progressDialog = new ProgressDialog(this);  
    progressDialog.setCancelable(false);  
    progressDialog.setIcon(R.mipmap.ic_launcher);  
    progressDialog.setTitle("Pobieram dane...");  
    progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);  
  
    progressDialog.show();  
    new Thread(new Runnable(){  
        public void run(){  
            for (int i=1; i<=10; i++) {  
                try {  
                    Thread.sleep(1500);  
                    progressDialog.incrementProgressBy(10);  
                } catch (InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
            progressDialog.dismiss();  
        }  
    }).start();  
}
```



Dodawanie przycisków

```
progressDialog.setButton(DialogInterface.BUTTON_POSITIVE, "OK",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,
            int which)
        {
            Toast.makeText(getBaseContext(), "Kliknięto OK",
                Toast.LENGTH_SHORT).show();
        }
    });
progressDialog.setButton(DialogInterface.BUTTON_NEGATIVE,
    "Anuluj",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,
            int which)
        {
            Toast.makeText(getBaseContext(),
                "Kliknięto Anuluj",
                Toast.LENGTH_SHORT).show();
        }
    });
```

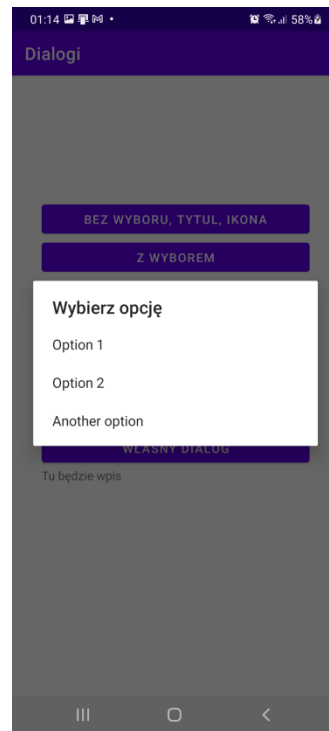
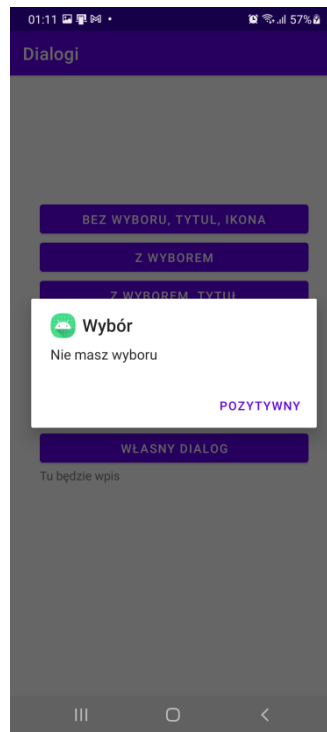
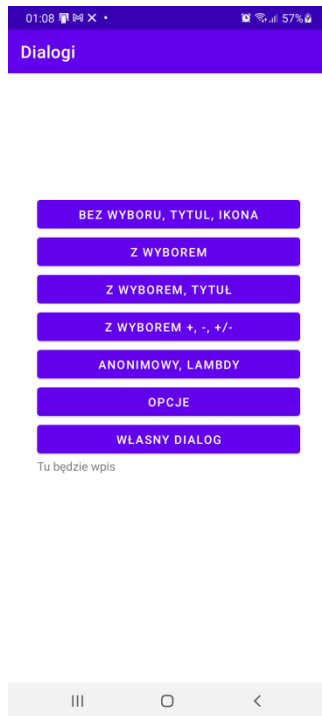
<https://developer.android.com/reference/android/app/ProgressDialog.html>



AlertDialog

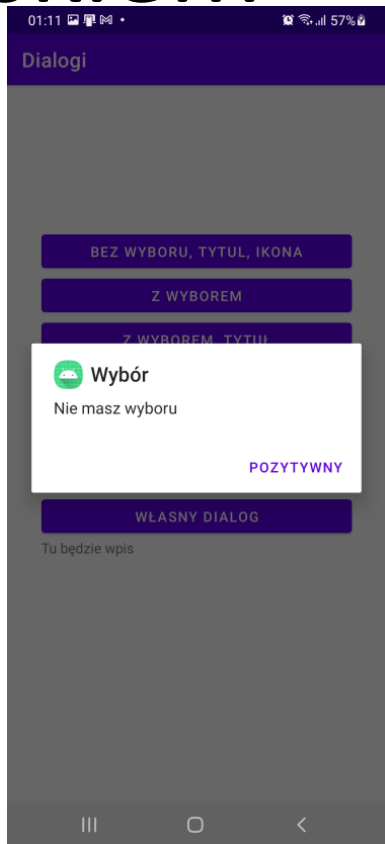
- Wyświetla małe okno pływające, znajdujące się nad aktualnym widokiem.
- Prezentuje użytkownikowi komunikat wraz z opcjonalnymi przyciskami.
- Dialog jest zamykany poprzez kliknięcie danego przycisku bądź dotknięcie obszaru poza dialogiem.

Program Dialogi



Prosty dialog z 1. przyciskiem

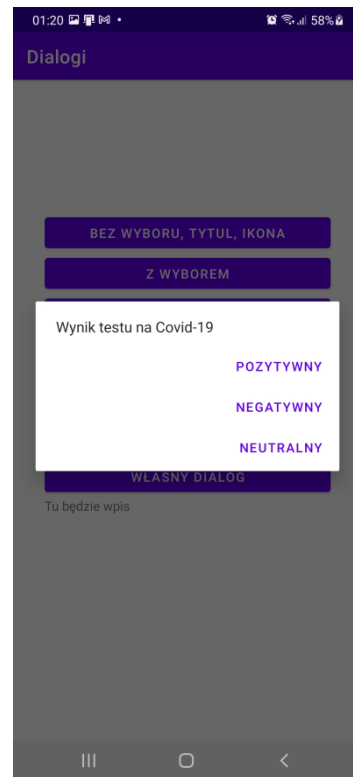
```
public void onClickButton0(View view) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(this);  
    builder.setIcon(R.mipmap.ic_launcher);  
    builder.setTitle("Wybór");  
    builder.setMessage("Nie masz wyboru");  
    builder.setPositiveButton(  
        "Pozytywny",  
        new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int id) {  
                Toast.makeText(getApplicationContext(), "Pozytywny",  
                    Toast.LENGTH_SHORT).show();  
                dialog.cancel();  
            }  
        });  
    AlertDialog alertDialog = builder.create();  
    alertDialog.show();  
}
```



Prosty dialog z 3. przyciskami

```
public void onClickButton1(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Wynik testu na Covid-19");

    builder.setPositiveButton(
        "Pozytywny",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(getApplicationContext(), "Pozytywny", Toast.LENGTH_SHORT).show();
                dialog.cancel();
            }
        });
    builder.setNegativeButton("Negatywny", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Negatywny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
    builder.setNeutralButton("Neutralny", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Neutralny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
    builder.create().show();
}
```



Prosty dialog z 3. przyciskami i tytułem

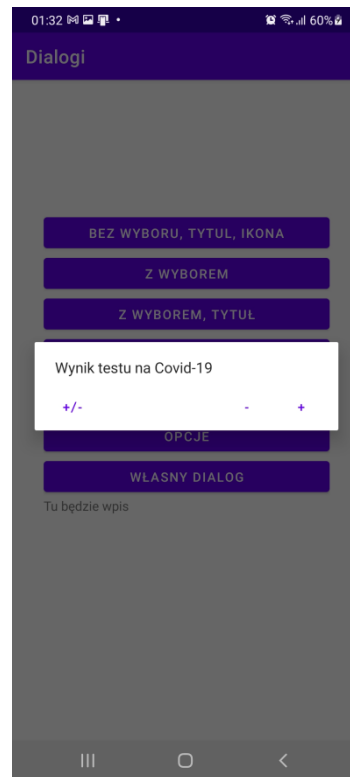
```
public void onClickButton2(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Wynik testu na Covid-19");
    builder.setTitle("Tytuł");
    builder.setNeutralButton("Neutralny", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Neutralny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
    builder.setPositiveButton(
        "Pozytywny",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(getApplicationContext(), "Pozytywny", Toast.LENGTH_SHORT).show();
                dialog.cancel();
            }
        }
    );
    builder.setNegativeButton("Negatywny", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Negatywny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
    builder.create().show();
}
```



Przyciski zdefiniowane w innej kolejności niż poprzednio

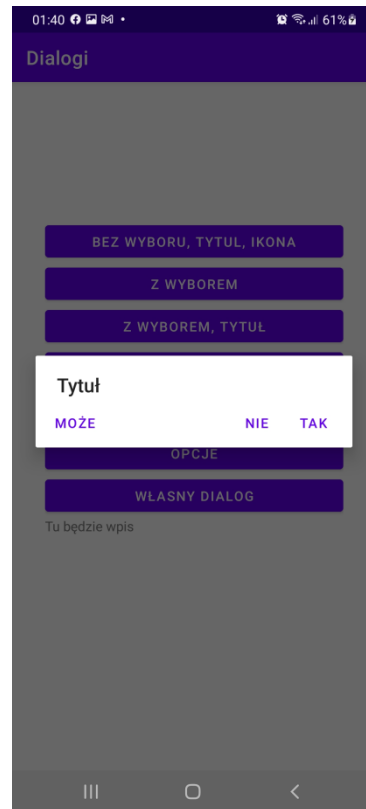
Prosty dialog z 3. i krótkimi napisami

```
public void onClickButton3(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Wynik testu na Covid-19");
    builder.setNeutralButton("+/-", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Neutralny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
    builder.setPositiveButton(
        "+",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(getApplicationContext(), "Pozytywny", Toast.LENGTH_SHORT).show();
                dialog.cancel();
            }
        }
    );
    builder.setNegativeButton("-", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Negatywny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
    builder.create().show();
}
```



Dialog anonimowy i lambda

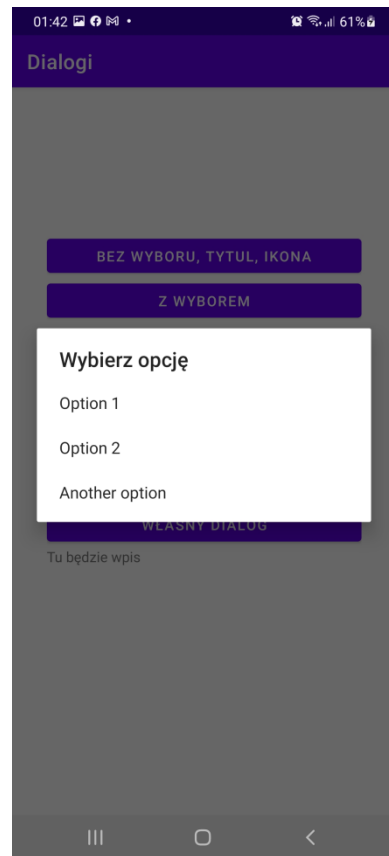
```
public void onClickButton4(View view) {
    new AlertDialog.Builder(this)
        .setTitle("Tytuł")
        .setNegativeButton("Nie", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getApplicationContext(), "Negatywny", Toast.LENGTH_SHORT).show();
                dialog.cancel();
            }
        })
        .setNeutralButton("Może", (dialog, which) -> {
            Toast.makeText(getApplicationContext(), "Neutralny", Toast.LENGTH_SHORT).show();
            dialog.cancel();
        })
        .setPositiveButton("Tak", (dialog, which) -> {
            Toast.makeText(getApplicationContext(), "Pozytywny: " + which, Toast.LENGTH_SHORT).show();
            dialog.cancel();
        })
        .create().show();
}
```



Dialog z wyborem opcji

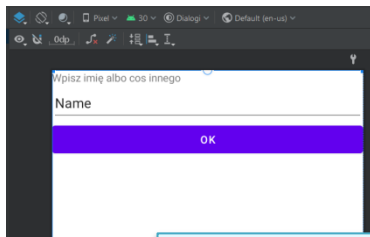
```
public void onClickButton7(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Wybierz opcję");

    final String[] options = {"Option 1", "Option 2", "Another option"};
    builder.setItems(options, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "pozycja =" + which, Toast.LENGTH_SHORT).show();
            //dialog.cancel();
        }
    }).create().show();
}
```



Dialog z zaprojektowanym układem

```
Context activityContext;  
TextView txtMsg;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    activityContext = this;  
    txtMsg = (TextView)findViewById(R.id.textView2) ;  
}
```



Nowy layout dialog

```
public void onClickButton6(View view) {  
    final Dialog customDialog = new Dialog(activityContext);  
    customDialog.setTitle("Custom Dialog Title");  
    customDialog.setContentView(R.layout.dialog);  
    final EditText sd_txtInputData = (EditText) customDialog  
        .findViewById(R.id.editTextUsername);  
    ((Button) customDialog.findViewById(R.id.button77))  
        .setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                txtMsg.setText(sd_txtInputData.getText().toString());  
                Toast.makeText(activityContext, sd_txtInputData.getText().toString(),  
                    Toast.LENGTH_SHORT).show();  
                customDialog.dismiss();  
            }  
        });  
    customDialog.show();  
}
```

