Android Studio - Java: Konwerter Jednostek

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

Ogólne

Aplikacja przelicza jedną jednostkę na inną (np. metry na mile). Wersja rozbudowana składa się z trzech *Activities*, kolorów definiowanych w „*colors.xml*" oraz tekstów definiowanych w „*strings.xml*".

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 - wersja podstawowa

Pierwsze Activity, plik „activity_main.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF0000"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/layoutLogoImage"
        android:layout_width="273dp"
        android:layout_height="166dp"
        android:adjustViewBounds="true"
        android:contentDescription="Logo aplikacji"
        android:src="@drawable/logo"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintVertical_bias="0.4" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Pierwsze Activity, plik „MainActivity.java”

```
package com.example.nazwaAplikacji;
// import ...
public class MainActivity extends AppCompatActivity {
  private final Handler harmonogramZadan = new Handler(Looper.getMainLooper());
  private final Runnable przejscieDoMenu = () -> {
    Intent zamiarOtwarciaSercaAplikacji = new Intent(MainActivity.this,
    KonwerterJednostek1.class);
    startActivity(zamiarOtwarciaSercaAplikacji);
    finish();
  };

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
      Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
      v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
      return insets;
    });

    ImageView logo = findViewById(R.id.layoutLogoImage);
    logo.setOnClickListener(kliknietyObiektLogo -> {
      goToWWW();
    });

    harmonogramZadan.postDelayed(przejscieDoMenu, 4000);
  }
```

```
    private void goToWWW() {
        harmonogramZadan.removeCallbacks(przejscieDoMenu);
        Intent zamiarOtwarciaWWW = new Intent(Intent.ACTION_VIEW,
        Uri.parse("https://3bird.pl"));
        startActivity(zamiarOtwarciaWWW);
        finish();
    }

    @Override
    protected void onDestroy() {
        harmonogramZadan.removeCallbacks(przejscieDoMenu);
        super.onDestroy();
    }
}
```

Drugie Activity, plik „activity_konwerter_jednostek1.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/layoutKonwerterJednostek1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".KonwerterJednostek1">

    <TextView
        android:id="@+id/layoutTytulKonwertera"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Konwerter jednostek"
        android:textSize="28sp"
        android:textStyle="bold"
        android:layout_marginTop="40dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/layoutTekstInfo1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Wybierz najpierw kategorię"
    android:textSize="12sp"
    android:textStyle="normal"
    android:layout_marginTop="40dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutTytulKonwertera" />

<Button
    android:id="@+id/layoutButtonKategoria"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Kategoria"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutTekstInfo1" />

<TextView
    android:id="@+id/layoutTekstInfo2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Wybierz jednostkę"
    android:textSize="12sp"
    android:textStyle="normal"
    android:layout_marginTop="60dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutButtonKategoria" />

<LinearLayout
    android:id="@+id/layoutInput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutTekstInfo2">

    <Button
        android:id="@+id/layoutButtonJednostka1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="Jednostka" />

    <EditText
        android:id="@+id/layoutEditText1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:hint="0"
        android:inputType="numberDecimal|numberSigned" />
</LinearLayout>

<LinearLayout
    android:id="@+id/layoutOutput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="30dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutInput">

    <Button
      android:id="@+id/layoutButtonJednostka2"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="2"
      android:text="Jednostka" />

    <TextView
      android:id="@+id/layoutTekstWynikowy"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="2"
      android:text="Wynik..."
      android:textColor="#FF0000"
      android:textSize="18sp"
      android:padding="8dp"/>
  </LinearLayout>

  <Button
    android:id="@+id/layoutButtonPrzelicz"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="60dp"
    android:text="Przelicz"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutOutput" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Drugie Activity, plik „KonwerterJednostek1.java”

```
package com.example.nazwaAplikacji;
// import...
public class KonwerterJednostek1 extends AppCompatActivity {
    private EditText liczbaZ;
    private TextView liczbaDo;
    private Button jednostka1;
    private Button jednostka2;
    private Button kategoria;
    private int wybranaJednostka1 = -1;
    private int wybranaJednostka2 = -1;
    private int numerKategorii = -1;

    private final String[] kategorie = {
            "Temperatura",
            "Odległość",
            "Powierzchnia"
    };

    private final String[][] jednostki = {
            {"Celsius", "Fahrenheit", "Kelvin"},
            {"Metry", "Kilometry", "Mile"},
            {"m²", "ha", "km²"}
    };

    private final double[][] wartosci = {
            {},
            {1, 1000, 1609.344},
            {1, 10000, 1000000}
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_konwerter_jednostek1);
```

```
    liczbaZ = findViewById(R.id.layoutEditText1);
    liczbaDo = findViewById(R.id.layoutTekstWynikowy);
    jednostka1 = findViewById(R.id.layoutButtonJednostka1);
    jednostka2 = findViewById(R.id.layoutButtonJednostka2);
    kategoria = findViewById(R.id.layoutButtonKategoria);
    Button przelicz = findViewById(R.id.layoutButtonPrzelicz);

    kategoria.setOnClickListener(kliknietyPrzycisk -> wybierzKategorie());
    jednostka1.setOnClickListener(kliknietyPrzycisk -> wybierzJednostke(
            jednostka1, true));
    jednostka2.setOnClickListener(kliknietyPrzycisk -> wybierzJednostke(
            jednostka2, false));
    przelicz.setOnClickListener(kliknietyPrzycisk -> przelicz());
}

private void wybierzKategorie() {
    new AlertDialog.Builder(this)
            .setTitle("Wybierz kategorię")
            .setItems(kategorie, (dialog, opcja) -> {
                numerKategorii = opcja;
                kategoria.setText(kategorie[opcja]);
                wybranaJednostka1 = -1;
                wybranaJednostka2 = -1;
                jednostka1.setText("Jednostka");
                jednostka2.setText("Jednostka");
            })
            .show();
}

private void wybierzJednostke(Button numerPrzyciskuJednostki, boolean czyJednostka1) {
    if (numerKategorii == -1) {
        Toast.makeText(this, "Wybierz najpierw kategorię", Toast.LENGTH_SHORT).show();
        return;
    }
```

```
    new AlertDialog.Builder(this)
            .setTitle("Wybierz jednostkę")
            .setItems(jednostki[numerKategorii], (dialog, opcja) -> {
                numerPrzyciskuJednostki.setText(jednostki[numerKategorii][opcja]);
                if (czyJednostka1) {
                    wybranaJednostka1 = opcja;
                }
                else {
                    wybranaJednostka2 = opcja;
                }
            })
            .show();
    }

    private void przelicz() {
        try {
            double liczba = Double.parseDouble(liczbaZ.getText().toString().replace(",", "."));
            if (wybranaJednostka1 == -1 || wybranaJednostka2 == -1) {
                Toast.makeText(this, "Wybierz jednostka", Toast.LENGTH_SHORT).show();
                return;
            }
            double wynik;
            if (numerKategorii == 0) {
                wynik = konwertujTemperature(liczba);
            }
            else {
                wynik = liczba * wartosci[numerKategorii][wybranaJednostka1];
                wynik = wynik / wartosci[numerKategorii][wybranaJednostka2];
            }
            liczbaDo.setText(String.format("%.4f", wynik));
        }
        catch (NumberFormatException e) {
            Toast.makeText(this, "Nieprawidłowa liczba", Toast.LENGTH_SHORT).show();
        }
```

```
    }

private double konwertujTemperature(double liczba) {
    if (wybranaJednostka1 == 1) {
        liczba = (5.0 / 9.0) * (liczba - 32);
    }
    else if (wybranaJednostka1 == 2) {
        liczba = liczba - 273.15;
    }
    if (wybranaJednostka2 == 1) {
        liczba = 32 + (9.0 / 5.0) * liczba;
    }
    else if (wybranaJednostka2 == 2) {
        liczba = liczba + 273.15;
    }
    return liczba;
  }
}
```

Kod - wersja rozbudowana

Plik „strings.xml”

```
<resources>
  <string name="app_name">Java Konwerter Jednostek</string>
  <string name="logo_description">Logo aplikacji</string>
  <string name="copyright">(c) Copyright by 3bird Projects.</string>
  <string name="konwerter_jednostek">Konwerter jednostek</string>
  <string name="back_exit">Aby zamknąć aplikację, wciśnij ponownie przycisk Wstecz</string>
  <string name="kolejne_aplikacje">Tu pojawią się kolejne przyciski do innych
  aplikacji...</string>
  <string name="converter_title">Konwerter jednostek</string>
  <string name="select_category_first">Wybierz najpierw kategorię</string>
  <string name="unit">Jednostka</string>
  <string name="paste">Wklej</string>
  <string name="copy">Kopiuj</string>
```

```
    <string name="result">Wynik</string>
    <string name="calculate">Przelicz</string>
    <string name="select_unit">Wybierz jednostki...</string>
    <string name="category">Kategoria</string>
    <string name="wrong_number">Nieprawidłowa liczba</string>
    <string name="copied_result">Skopiowano wynik</string>
    <string name="empty_clipboard">Schowek jest pusty</string>
</resources>
```

Plik „colors.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="red">#FFFF0000</color>
    <color name="splash_background">#FFFF0000</color>
</resources>
```

Pierwsze Activity, plik „activity_main.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/splash_background"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/layoutLogoImage"
        android:layout_width="273dp"
        android:layout_height="166dp"
        android:adjustViewBounds="true"
```

```
    android:contentDescription="@string/logo_description"
    android:src="@drawable/logo_3bird"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintVertical_bias="0.4"/>

  <TextView
    android:id="@+id/layoutCompanyName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/copyright"
    android:textSize="16sp"
    android:textColor="@color/white"
    android:layout_marginBottom="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Pierwsze Activity, plik „MainActivity.java”

```
package com.example.nazwaAplikacji;
// import...
public class MainActivity extends AppCompatActivity {
  private MediaPlayer dzwiekStartowy;
  // Tworzymy uchwyt do głównego wątku. "Looper" to rodzaj harmonogramu w pętli
  // (przypomina crona).
  // Jego praca polega na nieustannym pobieraniu ze stosu kolejnych zadań do wykonania,
  // np. odśwież ekran po kliknięciu, przejdź do drugiego Activity po 4s, pokaż wynik, itd.
  // Handler to narzędzie do dodawania kolejnych zadań do Loopera.
  // Runnable to po prostu zadanie dodane do kolejki.
  private final Handler harmonogramZadan = new Handler(Looper.getMainLooper());
  // Definiujemy zadanie dla Loopera:
  private final Runnable przejscieDoMenu = () -> {
```

```
    Intent zamiarOtwarciaSercaAplikacji = new Intent(MainActivity.this, SerceAplikacji.class);
    startActivity(zamiarOtwarciaSercaAplikacji);
    finish();
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });

    ImageView logo = findViewById(R.id.layoutLogoImage);
    logo.setOnClickListener(kliknietyObiektLogo -> {
        // android.util.Log.d("LOGO", "Kliknięto logo"); // Dla celów diagnostycznych
        goToWWW();
    });

    // Odtwarzany dźwięk przy starcie aplikacji (wav umieszczamy w folderze app / src / main /
    // res / raw)
    dzwiekStartowy = MediaPlayer.create(this, R.raw.start);
    if (dzwiekStartowy != null) {
        // Po zakończeniu odtwarzania wykonaj poniższy kod:
        dzwiekStartowy.setOnCompletionListener(odtwarzacz -> {
            odtwarzacz.release(); // Zwolnij zasoby
            dzwiekStartowy = null; // Anulowanie obiektu z pamięci
        });
        dzwiekStartowy.start();
    }
    // Po 4 sekundach wyświetlania wizytówki, uruchom drugie Activity:
```

```
        harmonogramZadan.postDelayed(przejscieDoMenu, 4000);
    } // Koniec onCreate


    // Otwiera domyślną przeglądarkę internetową i wyświetla stronę www:
    private void goToWWW() {
        // Skoro przechodzimy do strony WWW, to usuwamy z kolejki Handlera
        // wcześniej zaplanowane zadanie "przejscieDoMenu".
        harmonogramZadan.removeCallbacks(przejscieDoMenu);
        Intent zamiarOtwarciaWWW = new Intent(Intent.ACTION_VIEW,
        Uri.parse("https://3bird.pl"));
        startActivity(zamiarOtwarciaWWW);
        finish();
    }

    // Usunięcie oczekujących zadań na wypadek, gdyby użytkownik nieoczekiwanie zakończył
    // aplikację:
    @Override
    protected void onDestroy() {
        // Na wszelki wypadek usuwamy wszystkie oczekujące zadania przejścia do innego Activity:
        harmonogramZadan.removeCallbacks(przejscieDoMenu);
        // Na wszelki wypadek usuwamy także obiekt MediaPlayer (aplikacja została zamknięta
        // przed zakończeniem odtwarzania):
        if (dzwiekStartowy != null) {
            dzwiekStartowy.release();
            dzwiekStartowy = null;
        }
        // Wywołanie metody klasy nadrzędnej:
        super.onDestroy();
    }
}
```

Drugie Activity, plik „activity_serce_aplikacji.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/layoutSerceAplikacji"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SerceAplikacji">

    <Button
        android:id="@+id/layoutButtonKonwerter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/konwerter_jednostek"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="50dp"/>

    <TextView
        android:id="@+id/layoutInfoText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/kolejne_aplikacje"
        android:textSize="14sp"
        android:layout_marginTop="50dp"
        android:gravity="center"
        app:layout_constraintTop_toBottomOf="@id/layoutButtonKonwerter"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Drugie Activity, plik „SerceAplikacji.java”

```
package com.example.nazwaAplikacji;
// import...
public class SerceAplikacji extends AppCompatActivity {
```

```
private static final String TAG = "SerceAplikacji";
private boolean oczekiwanieNaDrugieCofnij = false;
// Definiujemy nowy handler dla Loopera:
private final Handler harmonogramZadan = new Handler(Looper.getMainLooper());
// Przekazujemy Looperowi nowe zadanie do wykonania (wyzerowanie stanu oczekiwania
// na drugie naciśnięcie przycisku WSTECZ po 2 sekundach bezczynności):
private final Runnable resetBackState = () -> oczekiwanieNaDrugieCofnij = false;

@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  EdgeToEdge.enable(this);
  setContentView(R.layout.activity_serce_aplikacji);
  ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.layoutSerceAplikacji),
  (v, insets) -> {
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
    return insets;
  });

  Button buttonKonwerter = findViewById(R.id.layoutButtonKonwerter);
  buttonKonwerter.setOnClickListener(kliknietyObiektPrzycisku -> {
    Intent zamiarOtwarciaKonwerteraJednostek = new Intent(this,
    KonwerterJednostek1.class);
    startActivity(zamiarOtwarciaKonwerteraJednostek);
  });

  // Poniżej tworzymy własną obsługę przycisku BACK (dwa kliknięcia zamiast jednego).
  // Po pojedynczym kliknięciu BACK nie jest wykonywana domyślna funkcja "Zamknij
  // aplikację", lecz wykonywany jest wewnętrzny kod (oczekiwanie na drugie kliknięcie BACK).
  // Czyli "getOnBackPressedDispatcher()" przechwytuje pierwsze naciśnięcie BACK i dodaje
  // nową funkcję do obsługi kliknięcia BACK (addCallback()). Parametr "this" oznacza
  // bieżące Activity, a parametr "true" oznacza, że nowy przycisk jest od razu aktywny.
  getOnBackPressedDispatcher().addCallback(this, new OnBackPressedCallback(true) {
```

```
        @Override
        // Nadpisujemy domyślną metodę zamykania aplikacji w Android (dwa kliknięcia we
        // WSTECZ):
        public void handleOnBackPressed() {
          // Sprawdzamy, czy użytkownik nacisnął BACK po raz pierwszy:
          if (!oczekiwanieNaDrugieCofnij) {
            // Ustawiamy flagę informującą, że pierwsze naciśnięcie już nastąpiło:
            oczekiwanieNaDrugieCofnij = true;
            Toast.makeText(SerceAplikacji.this, R.string.back_exit,
            Toast.LENGTH_SHORT).show();
            // Umieszczamy w harmonogramieZadan opóźnione o 2 sekundy zadanie wyzerowania
            // flagi "oczekiwanieNaDrugieCofnij". Jeśli użytkownik nie naciśnie drugi raz BACK,
            // cała procedura zacznie się od początku.
            harmonogramZadan.postDelayed(resetBackState,2000);
          }
           else {
            // Gdy użytkownik jednak nacisnął druga raz BACK w ciągu 2 sekund:
            finishAffinity(); // W przeciwieństwie do finish(), usuwa cały stos aktywności aplikacji.
          }
          // Definiujemy sobie własny komunikat w logach, gdy użyta zostanie ta metoda:
          Log.d(TAG, "Wciśnięto przycisk BACK");
        }
      });
    } // Koniec onCreate()

    // Usunięcie oczekujących zadań w przypadku nagłego zamknięcia aplikacji:
    @Override
    protected void onDestroy() {
      harmonogramZadan.removeCallbacks(resetBackState);
      super.onDestroy();
    }
}
```

Trzecie Activity, plik „activity_konwerter_jednostek1.xml”

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/layoutKonwerterJednostek1"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="16dp"
  tools:context=".KonwerterJednostek1">

  <TextView
    android:id="@+id/layoutTytulKonwertera"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/converter_title"
    android:textSize="28sp"
    android:textStyle="bold"
    android:layout_marginTop="40dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/layoutTekstInfo1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/select_category_first"
    android:textSize="12sp"
    android:textStyle="normal"
    android:layout_marginTop="40dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutTytulKonwertera" />
```

```
<Button
    android:id="@+id/layoutButtonKategoria"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="@string/category"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutTekstInfo1" />

<TextView
    android:id="@+id/layoutTekstInfo2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/select_unit"
    android:textSize="12sp"
    android:textStyle="normal"
    android:layout_marginTop="60dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutButtonKategoria" />

<!-- Wiersz INPUT (wejściowy) -->
<LinearLayout
    android:id="@+id/layoutInput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutTekstInfo2">
```

```
    <Button
        android:id="@+id/layoutButtonJednostka1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="@string/unit" />

    <EditText
        android:id="@+id/layoutEditText1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:hint="0"
        android:inputType="numberDecimal|numberSigned" />

    <ImageButton
        android:id="@+id/layoutButtonWklej"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:contentDescription="@string/paste"
        android:src="@drawable/outline_content_paste_24" />
</LinearLayout>

<!-- Wiersz OUTPUT (wyjściowy) -->
<LinearLayout
    android:id="@+id/layoutOutput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutInput">
```

```
    <Button
        android:id="@+id/layoutButtonJednostka2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="@string/unit" />

    <TextView
        android:id="@+id/layoutTekstWynikowy"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="@string/result"
        android:textColor="@color/red"
        android:textSize="18sp"
        android:padding="8dp"/>

    <ImageButton
        android:id="@+id/layoutButtonKopiuj"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:contentDescription="@string/copy"
        android:src="@drawable/outline_file_copy_24" />
</LinearLayout>

<Button
    android:id="@+id/layoutButtonPrzelicz"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="60dp"
    android:text="@string/calculate"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@id/layoutOutput" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Trzecie Activity, plik „KonwerterJednostek1.java”

```
package com.example.nazwaAplikacji;
// import ...
public class KonwerterJednostek1 extends AppCompatActivity {
    private EditText liczbaZ;
    private TextView liczbaDo;
    private Button jednostka1;
    private Button jednostka2;
    private Button kategoria;
    private int wybranaJednostka1 = -1;
    private int wybranaJednostka2 = -1;
    private int numerKategorii = -1;

    // Tablica jednowymiarowa:
    private final String[] kategorie = {
            "Temperatura",          // Indeks 0
            "Odległość",            // Indeks 1
            "Powierzchnia",         // Indeks 2
            "Prędkość"              // Indeks 3
    };

    // Tablica dwuwymiarowa:
    private final String[][] jednostki = {
            {"Celsius", "Fahrenheit", "Kelvin"},    // Indeks [0][0,1,2]
            {"Metry", "Kilometry", "Mile"},         // Indeks [1][0,1,2]
            {"m²", "ha", "km²"},                    // Indeks [2][0,1,2]
            {"km/h", "mile/h", "m/s"}               // Indeks [3][0,1,2]
    };

    // Tablica dwuwymiarowa:
    private final double[][] wartosci = {
            {},                                     // Temperatura (osobna metoda)
```

```
        {1, 1000, 1609.344},          // Odległość, indeks [1][0,1,2]
        {1, 10000, 1000000},          // Powierzchnia, indeks [2][0,1,2]
        {1, 1.609344, 3.6}            // Prędkość, indeks [3][0,1,2]
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_konwerter_jednostek1);
    // Powiązanie zmiennych z obiektami View:
    liczbaZ = findViewById(R.id.layoutEditText1);
    liczbaDo = findViewById(R.id.layoutTekstWynikowy);
    jednostka1 = findViewById(R.id.layoutButtonJednostka1);
    jednostka2 = findViewById(R.id.layoutButtonJednostka2);
    kategoria = findViewById(R.id.layoutButtonKategoria);
    Button przelicz = findViewById(R.id.layoutButtonPrzelicz);
    ImageButton kopiuj = findViewById(R.id.layoutButtonKopiuj);
    ImageButton wklej = findViewById(R.id.layoutButtonWklej);

    // Nasłuchiwanie przycisków:
    kategoria.setOnClickListener(kliknietyPrzycisk -> wybierzKategorie());
    jednostka1.setOnClickListener(kliknietyPrzycisk -> wybierzJednostke(
            jednostka1, true));
    jednostka2.setOnClickListener(kliknietyPrzycisk -> wybierzJednostke(
            jednostka2, false));
    przelicz.setOnClickListener(kliknietyPrzycisk -> przelicz());
    kopiuj.setOnClickListener(kliknietyPrzycisk -> kopiuj());
    wklej.setOnClickListener(kliknietyPrzycisk -> wklej());
} // Koniec onCreate

private void wybierzKategorie() {
    // Wyskakujące okienko wyboru kategorii; this to bieżące Activity:
    new AlertDialog.Builder(this)
            .setTitle("Wybierz kategorię")
```

```
        // Pierwszy parametr pobiera zawartość tablicy "kategorie[]".
        // Parametr "opcja" to numer kliknięty elementu [0,1,2,3] przypisany następnie
        // do zmiennej "numerKategorii". Parametr "dialog" reprezentuje "okno dialogowe".
        .setItems(kategorie, (dialog, opcja) -> {
            numerKategorii = opcja;
            // Zmieniamy napis przycisku na wybraną kategorię:
            kategoria.setText(kategorie[opcja]);
            // Reset poprzednio wybranych kategorii (aby nie wskazywało na [0,1,2,3]):
            wybranaJednostka1 = -1;
            wybranaJednostka2 = -1; // Reset poprzednio wybranych kategorii
            jednostka1.setText(R.string.unit); // Przywracamy domyślny napis na przycisku
            jednostka2.setText(R.string.unit); // Przywracamy domyślny napis na przycisku
        })
        .show();
}

// Dwa parametry: który przycisk (jednostka1 czy jednostka2) i czy wartość tego przycisku
// jest "true" czy "false" (te wartości boolowskie są wcześniej celowo przypisane).
// Przycisk "jednostka1" ma celowo przypisaną wartość "true", a przycisk "jednostka2" ma
// celowo przypisaną wartość "false", aby za pomocą jednej metody obsłużyć dwa przyciski.
private void wybierzJednostke(Button numerPrzyciskuJednostki, boolean czyJednostka1) {
    // Jeśli użytkownik chce wybrać jednostkę, ale jeszcze nie wybrał żadnej kategorii:
    if (numerKategorii == -1) {
        Toast.makeText(this, R.string.select_category_first, Toast.LENGTH_SHORT).show();
        return;
    }
    // Jeśli zaś wcześniej wybrał kategorię, to teraz otwiera się okno dialogowe:
    new AlertDialog.Builder(this)
        .setTitle("Wybierz jednostkę")
        .setItems(jednostki[numerKategorii], (dialog, opcja) -> {
            // Zmienia napis na przycisku na wybraną jednostkę:
            numerPrzyciskuJednostki.setText(jednostki[numerKategorii][opcja]);
            if (czyJednostka1) {
                // Opcja to numer indeksu. Wykona się podczas kliknięcia
```

```
            // na przycisk "Jednostka1" (bo jest true).
            wybranaJednostka1 = opcja;
          }
          else {
            // Wykona się podczas kliknięcia na przycisk "Jednostka2"
            // (bo ma wartość false, więc wykona się "else").
            wybranaJednostka2 = opcja;
          }
        })
        .show();
}

private void przelicz() {
  try {
    // Poniżej, pobieramy obiekt typu Editable, zamieniamy go na String,
    // a następnie parseDouble() zamienia tego Stringa na typ prosty double
    // (sama wartość liczbowa, która nie może być null; nie mylić z typem "Double",
    // który jest obiektem zawierającym wartość liczbową).
    double liczba = Double.parseDouble(liczbaZ.getText().toString().replace(",", "."));
    // Jeśli użytkownik nie wybrał jednostki (czyli indeksu [0,1,2]):
    if (wybranaJednostka1 == -1 || wybranaJednostka2 == -1) {
      Toast.makeText(this, R.string.select_unit, Toast.LENGTH_SHORT).show();
      return;
    }

    double wynik; // Typ prosty double, czyli sama wartość liczbowa
    if (numerKategorii == 0) { // Jeśli użytkownik wybrał kategorię temperatury, indeks [0]
      wynik = konwertujTemperature(liczba);
    }
    else { // Jeśli użytkownik wybrał jedną z pozostałych kategorii.
      // Na przykład: 5 mil na kilometry.
      // Mile --> {2} (przelicznik jest na drugim indeksie licząc od 0, czyli trzeci w kolejności)
      // Kilometry --> {1} (przelicznik jest na pierwszym indeksie licząc od 0, czyli drugi w
      // kolejności)
```

```
            // Poniżej będzie więc tak:
            // wynik = 5mil * 1609.34;
            // wynik = 8046.72 / 1000 (czyli 8.04672km);
            // Bardziej konkretnie:
            // wynik = 5 * [1][2] (drugi wiersz, trzeci element, czyli 1609.34)
            // wynik = 8046.72 / [1][1] (drugi wiersz, drugi element, czyli 1000), bo {1, 1000,
            // 1609.344}
            wynik = liczba * wartosci[numerKategorii][wybranaJednostka1];
            wynik = wynik / wartosci[numerKategorii][wybranaJednostka2];
        }

        // W tej formie będzie bez zaokrąglania: liczbaDo.setText(String.valueOf(wynik));
        // A tak z zaokrągleniem do czterech miejsc po przecinku:
        liczbaDo.setText(String.format("%.4f", wynik));
    }
    catch (NumberFormatException e) {
        Toast.makeText(this, R.string.wrong_number, Toast.LENGTH_SHORT).show();
    }
} // Koniec metody "przelicz()"

// Temperatura wymaga osobnej metody, ponieważ nie da się jej przeliczać zwykłym
// mnożeniem i dzieleniem przez stały współczynnik, tak jak robimy to dla odległości,
// powierzchni czy prędkości.
// Najpierw wszystkie jednostki przeliczane są na Celsjusze:
private double konwertujTemperature(double liczba) {
    if (wybranaJednostka1 == 1) { // Jeśli użytkownik wybrał Fahrenheita (czyli indeks [1])
        liczba = (5.0 / 9.0) * (liczba - 32); // Przeliczanie na Celsjusze
    }
    else if (wybranaJednostka1 == 2) { // Jeśli użytkownik wybrał Kelvina (czyli indeks [2])
        liczba = liczba - 273.15; // Przeliczanie na Celsjusze
    }
    // Teraz Celsjusze przeliczane są na jednostkę docelową:
    if (wybranaJednostka2 == 1) { // Jeśli użytkownik chce na Fahrenheity (indeks [1])
        liczba = 32 + (9.0 / 5.0) * liczba; // Wynik w Fahrenheit
```

```
    }
    else if (wybranaJednostka2 == 2) { // Jeśli użytkownik chce na Kelviny (indeks [2])
        liczba = liczba + 273.15; // Wynik w Kelvinach
    }
    return liczba;
} // Koniec konwertujTemperature

private void kopiuj() {
    ClipboardManager schowek = getSystemService(ClipboardManager.class);
    // Tworzymy obiekt schowka zawierający dane tekstowe. Parametr "Wynik" to etykieta,
    // opis danych. Drugi parametr pobiera tekst typu TextView z pola "liczbaDo"
    // i zamienia go na typ String.
    ClipData dane = ClipData.newPlainText("Wynik", liczbaDo.getText().toString());
    // Ustawia ten obiekt jako główną (aktualną) zawartość schowka (poprzednia zawartość
    // zostanie zastąpiona):
    schowek.setPrimaryClip(dane);
    Toast.makeText(this, R.string.copied_result, Toast.LENGTH_SHORT).show();
}

private void wklej() {
    ClipboardManager schowek = getSystemService(ClipboardManager.class);
    // Jeśli schowek nie jest pusty, a zawartość została ustawiona jako "PrimaryClip":
    if (schowek != null && schowek.hasPrimaryClip()) {
        ClipData dane = schowek.getPrimaryClip(); // Pobrał zawartość i przypisał do zmiennej
        // Czy schowek nie jest pusty i czy znajduje się tam co najmniej jeden element
        // (czy liczba elementów w schowku jest większa niż 0). Uwaga: To "0" to nie jest indeks
        // lecz po prostu ilość elementów w schowku.
        if (dane != null && dane.getItemCount() > 0) {
            // Poniżej pobiera ze schowka obiekt typu ClipData.Item o indeksie [0]
            // i wydobywa z niego tylko część tekstową:
            liczbaZ.setText(dane.getItemAt(0).getText());
        }
    }
    else {
```

```
            Toast.makeText(this, R.string.empty_clipboard, Toast.LENGTH_SHORT).show();
        }
    }
}
```

Ocena celująca

Uwaga: Aby otrzymać ocenę celującą, należy dodatkowo:

- wykonać cały projekt bez pomocy nauczyciela w ciągu pierwszego tygodnia;

- utworzyć własną ikonę aplikacji;

- ustawić nazwę aplikacji na „*Java Konwerter Jednostek*" (nazwa znajdująca się pod ikoną aplikacji w smartfonie).

Ostatnia aktualizacja: 28 maja 2026.