|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Politechnika Bydgoska im. Jana i Jędrzeja Śniadeckich  Wydział Telekomunikacji, Informatyki i Elektrotechniki  al. prof. S. Kaliskiego 7, 85-796 Bydgoszcz | | | |  | |
| Przedmiot | **Programowanie urządzeń mobilnych** | | | Kierunek/Tryb |  |
| Nr laboratorium | 1 | Data wykonania |  | Grupa |  |
| Ocena |  | Data oddania |  | Imię Nazwisko |  |
| Nazwa ćwiczenia | Interfejs użytkownika EditText, Button i TextView | | |

**Cel ćwiczenia laboratoryjnego**

Celem ćwiczenia jest zapoznanie studentów z tworzeniem prostej aplikacji mobilnej na platformie Android przy użyciu interfejsu użytkownika **EditText**, **Button** i **TextView**. Studenci mają nauczyć się:

1. Projektować interfejs użytkownika, który obejmuje wprowadzanie danych, przycisk do przeliczenia oraz wyświetlanie wyników.
2. Obsługiwać zdarzenia naciśnięcia przycisku w celu przeliczenia danych.
3. Wykonywać proste obliczenia na danych wejściowych.
4. Wyświetlać wynik na ekranie w czytelny sposób.
5. Obsługiwać potencjalne błędy podczas wprowadzania danych.

**Zadanie**

Studenci zostaną poproszeni o stworzenie aplikacji mobilnej na platformie Android, która umożliwia użytkownikowi wprowadzenie temperatury w stopniach Celsjusza i przeliczenie jej na stopnie Fahrenheita. Aplikacja powinna obsługiwać sytuacje, w których użytkownik wprowadza niepoprawne dane oraz prezentować wynik w czytelny sposób na ekranie.

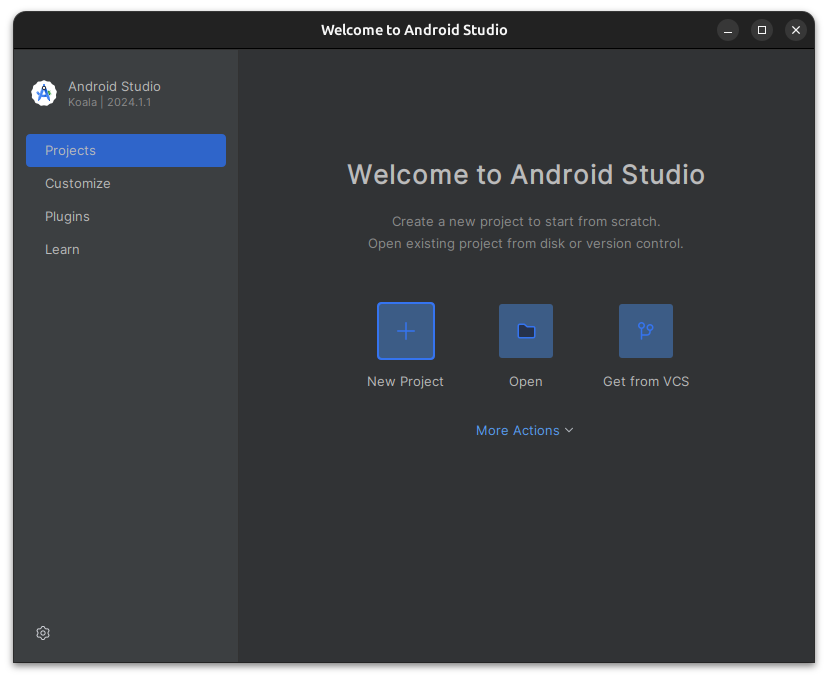
**Oczekiwane wyniki**

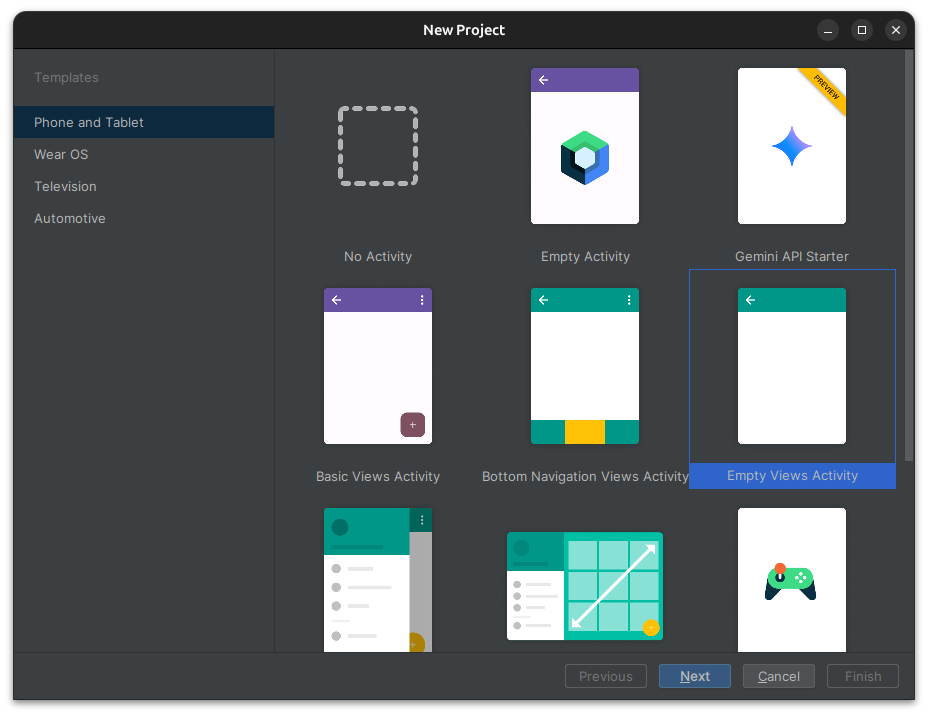
Po zakończeniu tego ćwiczenia studenci powinni być w stanie:

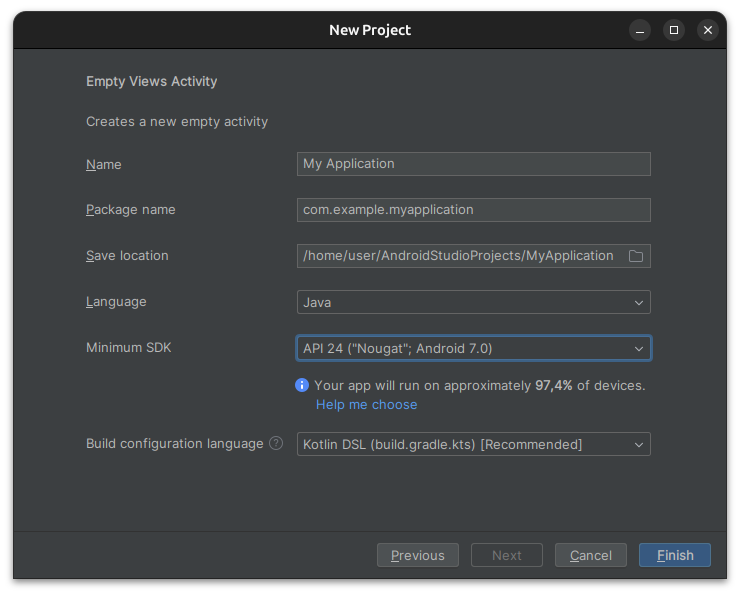
1. Tworzyć interfejsy użytkownika w środowisku Android.
2. Programować obsługę zdarzeń, takich jak naciśnięcie przycisku.
3. Przeprowadzać proste obliczenia na danych wejściowych.
4. Wyświetlać wyniki obliczeń na ekranie w formie tekstu.
5. Rozumieć i obsługiwać błędy wprowadzania danych.

**Przykład 1**

Utworzenie okna z nagłówkiem i tekstem powitania. Utwórz nowy projekt Android Studio: Rozpocznij od stworzenia nowego projektu Android Studio, wybierając odpowiednią konfigurację (np. "Empty Activity"). Upewnij się, że projekt jest skonfigurowany do pracy z językiem Java lub Kotlin, w zależności od twoich preferencji.





****

Zanim przejdziemy do modyfikacji kodu wygenerowanego przez kreator możemy uruchomić aplikację celem sprawdzenia działania środowiska programistycznego (SHIFT+F10). Jeżeli palikcaja się uruchomi to przechodzimy do modyfikacji kodu:

Jedną zmetod zmiany zmiany nazwy nagłówka, będzie modyfikacja zmiennej app\_name w strings.xml

<resources>

<string name="app\_name">To jest moja nazwa aplikacji</string>

</resources>

Dodanie tekstu powitania

Edycja pliku strings.xml

<resources>

<string name="app\_name">Zadanie1</string>

<string name="powitanie">Dzień Dobry</string>

</resources>

Edycja pliku 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"

tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/powitanie"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

Zmiana wielkości tekstu powitania

Zmiany wielkości czcionki dokonamy przez modyfikację parametrów

Edycja pliku activity\_main.xml

android:textSize="30sp"

android:textStyle="bold"

lub modyfikując odpowiednie parametry textApperance w Common Atributes edytora komponentu TextView.

**Przykład 2**

Utworzenie aplikacji, która wyświetla powitanie (w losowo) wybranym języku. Utwórz nowy projekt Android Studio: Rozpocznij od stworzenia nowego projektu Android Studio, wybierając odpowiednią konfigurację (np. "Empty Views Activity"). Upewnij się, że projekt jest skonfigurowany do pracy z językiem Java lub Kotlin, w zależności od twoich preferencji.

Obraz zawierający zrzut ekranu, tekst, design

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Edycja pliku MainActivity.java

package com.example.zadanie2;

import android.os.Bundle;

import android.widget.TextView;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.graphics.Insets;

import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EdgeToEdge.enable(this);

setContentView(R.layout.activity\_main);

final TextView textView;

textView = findViewById(R.id.textView);

final String[] POWITANIE = {"Dzień dobry","Good morning","Guten morgen","Bonjour","Buenos dijas"};

Random r = new Random (System.currentTimeMillis());

int liczba = Math.abs(r.nextInt()%5);

String tekst = POWITANIE[liczba];

textView.setText(tekst);

System.out.println(tekst);

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;

});

}

}

Edycja pliku 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"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**Przykład 3**

Utwórz nowy projekt Android Studio: Rozpocznij od stworzenia nowego projektu Android Studio, wybierając odpowiednią konfigurację (np. "Empty Views Activity"). Upewnij się, że projekt jest skonfigurowany do pracy z językiem Java lub Kotlin, w zależności od twoich preferencji.

Edycja pliku layout: Otwórz plik activity\_main.xml w folderze res/layout i zdefiniuj interfejs użytkownika, dodając EditText, Button i TextView.

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

<RelativeLayout 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"

tools:context=".MainActivity">

<EditText

android:id="@+id/editText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Wprowadź tekst" />

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/editText"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="16dp"

android:text="Pokaż tekst" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/button"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="16dp"

android:text="" />

</RelativeLayout>

Kod w MainActivity: Otwórz plik MainActivity.java i zaimplementuj logikę obsługi interfejsu użytkownika. Przykład w języku Java:

package com.example.zadanie3;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.graphics.Insets;

import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

private EditText editText;

private Button button;

private TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EdgeToEdge.enable(this);

setContentView(R.layout.activity\_main);

editText = findViewById(R.id.editText);

button = findViewById(R.id.button);

textView = findViewById(R.id.textView);

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String enteredText = editText.getText().toString();

textView.setText(enteredText);

}

});

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;

});

}

}

Uruchomienie aplikacji: Uruchom aplikację na emulatorze lub na fizycznym urządzeniu Android. Po wpisaniu tekstu w EditText i naciśnięciu przycisku Button, wprowadzony tekst zostanie wyświetlony w TextView.

**Zadanie**

Należy utworzyć aplikację przeliczającą stopnie Celsjusza na stopnie Fahrenheita, na przykład na podstawie powyższego przykładu. W tym celu można dodać dodatkowe pole **EditText** do wprowadzania stopni Celsjusza, przycisk do przeliczenia oraz **TextView** do wyświetlenia wyniku w stopniach Fahrenheita. Uwzględnij możliwość podawania ujemnych wartości temperatury. Możesz użyć parseDouble() do zamiany ciągu tekstowego na wartość numeryczną. Zabezpiecz aplikację przed podaniem np. pustej wartości.

**Sprawozdanie**

Sprawozdanie z ćwiczenia w ramach nauki programowania aplikacji mobilnych powinno zawierać istotne informacje i dokumentację dotyczącą zadania oraz jego realizacji. Oto kilka kluczowych elementów, które powinny być uwzględnione w sprawozdaniu:

1. **Tytuł i informacje ogólne**
   * Tytuł ćwiczenia.
   * Imię i nazwisko studenta.
   * Data realizacji ćwiczenia.
2. **Cel ćwiczenia**
   * Krótka informacja o celu i znaczeniu ćwiczenia, jak również o tym, czego studenci mieli się nauczyć.
3. **Opis projektu**
   * Opis funkcjonalności projektowanej aplikacji (przeliczanie stopni Celsjusza na Fahrenheita).
   * Krótka charakteryzacja interfejsu użytkownika, jakie widoki i elementy zostały użyte.
4. **Implementacja**
   * Opis procesu tworzenia projektu.
   * Omówienie wykorzystanych narzędzi i technologii (Android Studio, język Java/Kotlin).
   * Przedstawienie kodu źródłowego aplikacji, zarówno XML (layout) jak i kodu Java/Kotlin.
5. **Funkcje** 
   * Omówienie kluczowych funkcji aplikacji, takich jak obsługa przycisku, przeliczanie temperatury, obsługa błędów.
6. **Testowanie**
   * Opis testowania aplikacji, w tym przykłady testów, jakie przeprowadzono.
   * Raport z wynikami testów.
7. **Wyniki**
   * Przedstawienie wyników działania aplikacji.
   * Włączanie zrzutów ekranu lub opis demonstrujący, jak aplikacja działa.
8. **Podsumowanie**
   * Krótka ocena projektu i osiągnięć.
   * Wnioski wynikające z ćwiczenia.
9. **Trudności i błędy**
   * Informacje na temat ewentualnych problemów, na jakie napotkano oraz jak zostały one rozwiązane.
10. **Źródła i odniesienia**
    * Jeśli korzystano z materiałów lub źródeł zewnętrznych, warto je tu uwzględnić.
11. **Dodatkowe materiały**
    * Ewentualne dodatkowe materiały, takie jak kody źródłowe, zrzuty ekranu itp.

Warto zachować przejrzystość i czytelność w sprawozdaniu, używać odpowiednich nagłówków, numeracji stron, a także opisywać kroki projektowania i implementacji w sposób logiczny. Sprawozdanie powinno być dostatecznie szczegółowe, aby inny programista mógł zrozumieć projekt i ewentualnie go udoskonalić lub wykorzystać w przyszłości.