

Instrukcja do laboratorium z przedmiotu: Projektowanie Aplikacji Mobilnych

Warunkiem zaliczenia laboratorium jest oddanie następujących programów:

1. Kalkulator
2. Aplikacja pogodowa

Terminy oddawania programów:

Zjazd:	1	2	3	4	5	6	7	8	9	10
IO:						Kalkulator			Pogoda	
TI:							Kalkulator			Pogoda

W przypadku oddania programu “po terminie”, każdy kolejny termin obniża ocenę o 1 (np. oddanie programu kalkulator na zajęciach nr. 7 przez osobę z grupy IO daje możliwość uzyskania maksymalnie oceny dobrej (4) etc.).

Ocena z laboratorium będzie średnią arytmetyczna ocen uzyskanych z każdego oddawanego programu.

Program musi zostać napisany na platformę Android w środowisku **Android Studio** (nie ma możliwości korzystania z innych środowisk) w języku **Java** lub **Kotlin**.

Program musi pracować na urządzeniach typu:

- Telefon
- Tablet

W przypadku korzystania z własnych urządzeń, na pierwszych zajęciach należy sprawdzić aktualną wersję systemu i dostosować do niej swoje przyszłe programy.

Programy oddawane na zajęciach będą testowane na w/w. typach urządzeń, zarówno w orientacji pionowej jak i poziomej.

Istnieje możliwość pracy na własnych laptopach.

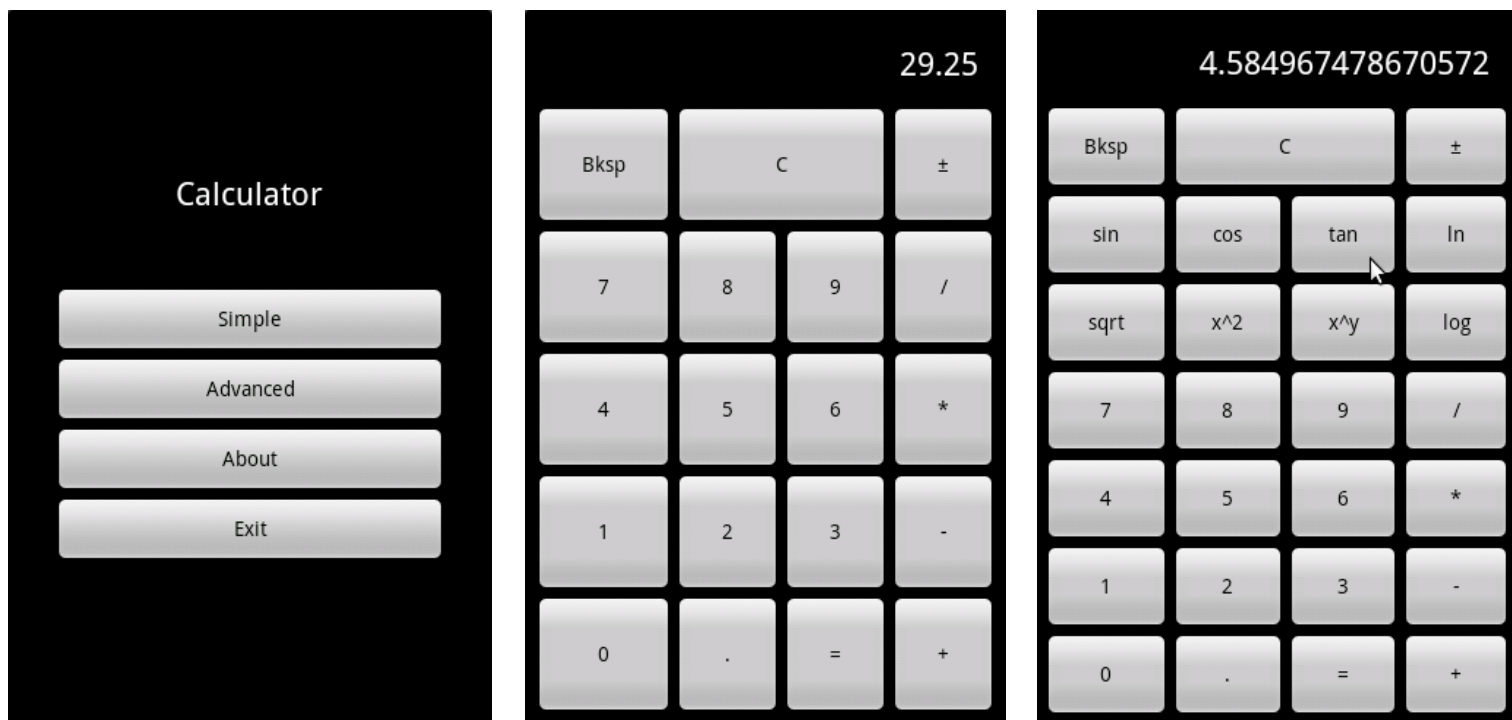
Ćwiczenie 1: Kalkulator

Celem ćwiczenia jest zapoznanie się z podstawowymi elementami interfejsu użytkownika (layouty, przyciski, pola tekstowe) oraz sposobem tworzenia aktywności i przełączania między nimi.

Założenia:

- 1) Aplikacja powinna zapewniać funkcjonalność kalkulatora, który w zależności od wyboru użytkownika pozwoli na wykonanie podstawowych lub zaawansowanych obliczeń.
- 2) Ekran startowy aplikacji powinien umożliwiać:
 - Wybór typu kalkulatora – prosty lub zaawansowany.
 - Wyświetlenie informacji o aplikacji (np. nazwa aplikacji, nazwisko autora, krótki opis).
 - Zamknięcie aplikacji.
- 3) Kalkulator w wersji prostej powinien umożliwiać wykonanie jedynie podstawowych operacji matematycznych (zmiennoprzecinkowych) w tym obowiązkowo:
 - Cztery podstawowe działania: dodawanie (+), odejmowanie(-), mnożenie(*), dzielenie(/).
 - Zmiana znaku liczby (+/-) - znak zmienia się po, a nie przed wprowadzeniem liczby.
 - Clear enter/clear (C/CE) - pierwsze naciśnięcie tego klawisza kasuje liczbę na wyświetlaczu i umożliwia wprowadzenie innej, bez zakłócania toku obliczeń. Dwukrotne naciśnięcie zeruje wszystkie rejestry kalkulatora (chyba, że istnieje zaimplementowana funkcja pamięci).
 - All clear (AC) - klawisz całkowitego kasowania rejestrów (i opcjonalnie pamięci) kalkulatora.
 - Wyświetlanie wyniku działania (=).
- 4) Kalkulator w wersji zaawansowanej, oprócz podstawowych, powinien dodatkowo implementować bardziej zaawansowane operacje w tym obowiązkowo:
 - Funkcje trygonometryczne: sinus (sin), cosinus (cos), tangens (tan).
 - Funkcje logarytmiczne: logarytm naturalny (ln), logarytm (log).
 - Obliczanie procentów (%).
 - Pierwiastkowanie (sqrt).
 - Potęgowanie (x^2), (x^y).
- 5) Dane wprowadzane przez użytkownika powinny być weryfikowane. W przypadku błędnych danych użytkownik powinien zostać o tym poinformowany (można w tym celu wykorzystać np. powiadomienie Toast).
- 6) Kalkulator musi pracować poprawnie w orientacji pionowej i poziomej. Operacja wprowadzona w trybie pionowym, może zostać kontynuowana z trybie poziomym.
- 7) Niedozwolone jest korzystanie z zewnętrznych bibliotek do ewaluacji wyrażeń matematycznych.

8) Przykładowe zrzuty ekranów:



Ćwiczenie 2: Aplikacja pogodowa

Celem ćwiczenia jest zapoznanie się z następującymi zagadnieniami:

- Interfejs użytkownika bazujący na fragmentach.
- Komunikacja pomiędzy fragmentami i aktywnością.
- Dostosowanie interfejsu użytkownika do różnych ekranów.
- Usługi sieciowe dostępne w systemie Android.
- Parsowanie plików XML/JSON.
- Przechowywanie danych w pamięci urządzenia (np. w prywatnym katalogu aplikacji).
- Przechowywanie danych w bazie danych.

Założenia:

- 1) Niniejsze ćwiczenie zakłada opracowanie aplikacji pozwalającej m.in. na pobieranie i wyświetlanie informacji o warunkach pogodowych (bieżących i prognozowanych) dla lokalizacji wybranych przez użytkownika.
- 2) Dane pogodowe można pobrać korzystając z API udostępnianego przez serwis <https://openweathermap.org>. Dane można pobierać w postaci plików JSON lub XML. Szczegółowe informacje dla programistów chcących korzystać z plików udostępnianych w ramach portalu yahoo.com można znaleźć pod adresem: <https://openweathermap.org/api>
- 3) Aplikacja powinna umożliwiać użytkownikowi zdefiniowanie listy ulubionych lokalizacji, dla których będą pobierane dane pogodowe.
- 4) W momencie uruchomienia aplikacja powinna sprawdzać czy możliwe jest połączenie z internetem. Jeżeli tak, pobierane są aktualne informacje na temat pogody i zapisywane w pamięci telefonu (w prywatnym katalogu aplikacji).
- 5) Jeżeli żadne połączenie internetowe nie jest aktywne w chwili uruchomienia aplikacji, informacje na temat pogody powinny zostać wczytane z pliku, który został zapisany podczas ostatniego połączenia. Ponadto użytkownik powinien być poinformowany o tym, że dane mogą być nieaktualne, a do aktualizacji wymagane jest połączenie internetowe.

Uwaga: Warto zastanowić się czy konieczne jest pobieranie informacji z Internetu przy każdym uruchomieniu aplikacji. Wśród informacji dostarczanych w pliku XML znajduje się czas ostatniej aktualizacji danych. Można więc zdefiniować czas, przez który dane mogą być trzymane w pamięci podręcznej, bez konieczności odświeżania. Można również z góry założyć pewien czas, przez który dane nie będą odświeżane.

- 6) Dane pogodowe powinny być prezentowane w przejrzystej formie. W tym celu należy przygotować interfejs aplikacji bazujący na fragmentach, tj.:
 - Fragment 1 – podstawowe dane, tj.: nazwa miejscowości, współrzędne geograficzne, czas, temperatura, ciśnienie, opis i reprezentacja graficzna warunków pogodowych.
 - Fragment 2 – dane dodatkowe np.: informacje o sile i kierunku wiatru, wilgotności, widoczności.
 - Fragment 3 – prognoza pogody na nadchodzące dni.

Uwaga: Układ fragmentów na ekranie powinien być zależny od jego orientacji i rozdzielczości. W przypadku orientacji pionowej dla urządzenia typu telefon należy zastosować przewijanie fragmentów np. za pomocą klasy ViewPager.

7) W aplikacji powinno być dostępne menu, które umożliwi:

- Odświeżenie informacji z internetu na żądanie użytkownika.
- Ustawienie/zmianę lokalizacji, dla których pobierane są dane.
- Wybór jednostek miary stosowanych w aplikacji.

Uwaga!

W przypadku, gdy prowadzący będzie miał podejrzenia co do samodzielności napisanej aplikacji, może uznać zadanie za niezaliczone bez możliwości poprawy lub może wyznaczyć jako zadanie indywidualne napisanie alternatywnej aplikacji o podobnych wymaganiach co splagiatowana aplikacja. Termin i ocena oddawania alternatywnej aplikacji będą identyczne jak aplikacji podstawowej tj. w przypadku “wychwycenia” plagiatu / niesamodzielnie napisanej aplikacji, prowadzący wyznacza zakres nowej aplikacji. Student, będąc “przyłapanym” na niesamodzielnej pracy w dniu pierwszego terminu oddawania pracy zostanie poproszony o napisanie aplikacji alternatywnej. W kolejnym terminie automatycznie najwyższą oceną jaką będzie mógł otrzymać to ocena dobra (4) - zgodnie z systemem oceniania opisanym na stronie 1.