

## Zadanie 1.

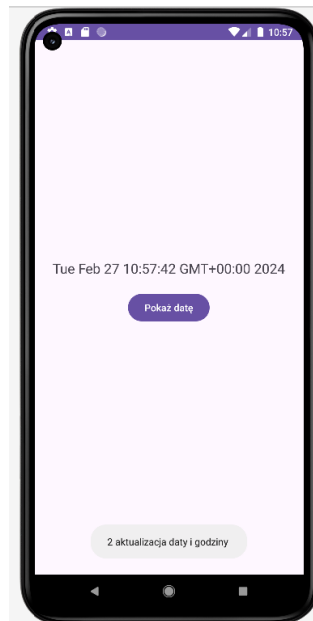
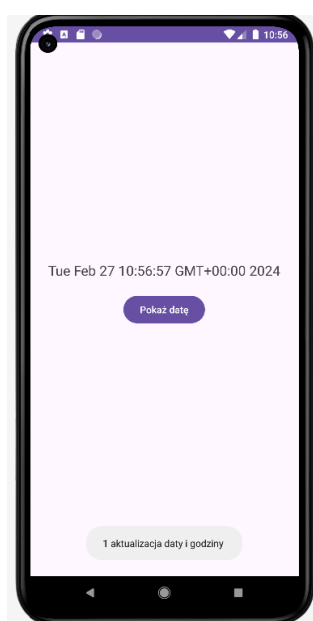
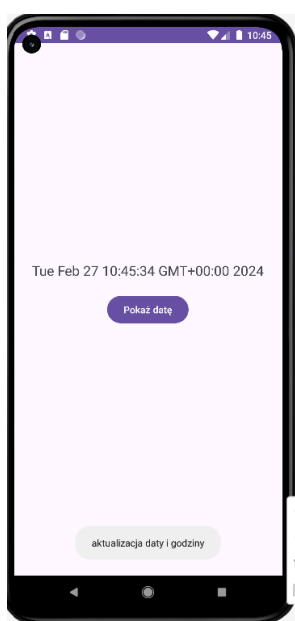
Stwórz aplikację mobilną w języku Java, która będzie wyświetlać aktualną datę i godzinę po naciśnięciu przycisku z krótką informacją w formie Toastu.

### 1. **TextView i Button:**

- Zweryfikuj, czy w pliku **activity\_main.xml** znajdują się elementy **TextView** i **Button** z odpowiednimi identyfikatorami (**textView** i **button**).
  - Upewnij się, że obiekty **text** i **btn** są poprawnie zainicjalizowane w metodzie **onCreate**.
2. Po uruchomieniu aplikacji w polu text wyświetla się bieżąca data i godzina.
  3. Ustaw listener dla przycisku, który po kliknięciu aktualizuje wyświetlaną datę i godzinę.
  4. Po kliknięciu przycisku wywołaj metodę **updateTime()**, która ustawia aktualną datę i godzinę w elemencie **TextView**.
  5. Jeśli jest to kolejne kliknięcie przycisku, wyświetl informację o aktualizacji razem z liczbą aktualizacji.
  6. Zdefiniuj licznik **i**, aby śledzić liczbę aktualizacji.
  7. Wykorzystaj komponent **Toast** do wyświetlania informacji na ekranie.
  8. Sprawdź, czy komponent **Toast** wyświetla się poprawnie, a także czy licznik **i** zlicza poprawnie kliknięcia przycisku.

### Aktualizacja Daty i Godziny:

```
private void updateTime() {  
    text.setText(new Date().toString());  
}
```



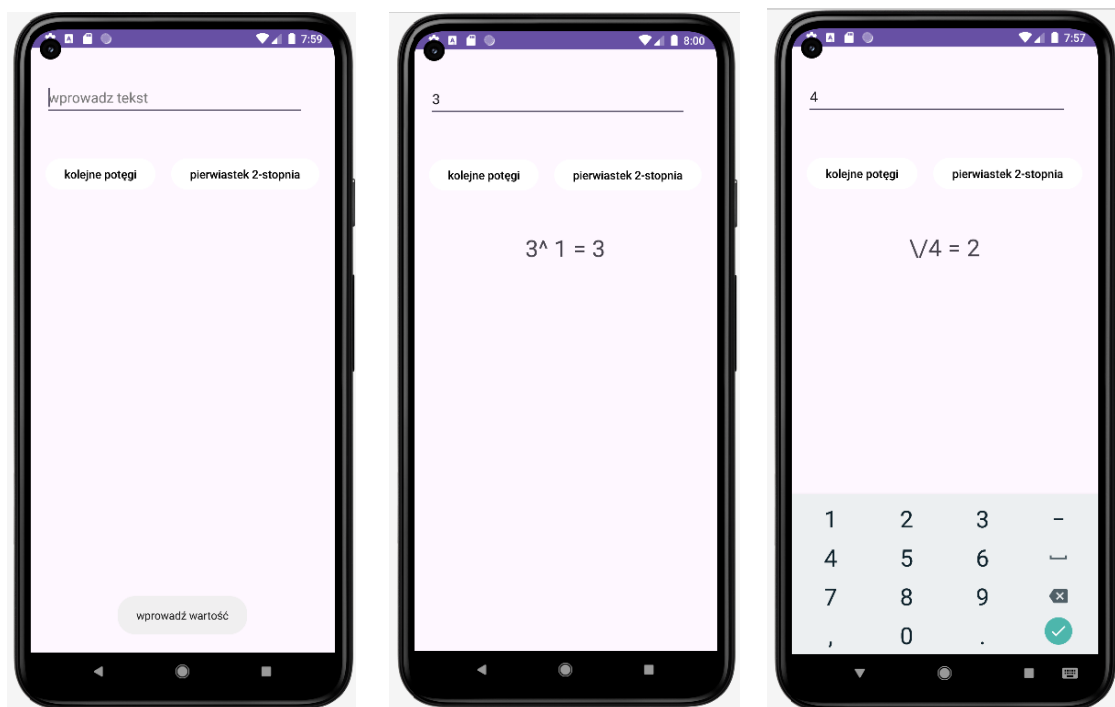
## Zadanie 2 Aplikacja potęgowanie i pierwiastek kwadratowy.

Aplikacja ta ma obsługiwać dwie operacje matematyczne: podnoszenie do potęgi i obliczanie pierwiastka kwadratowego.

Interfejs użytkownika powinien zawierać pole tekstowe do wprowadzania liczby, dwa przyciski ("btn1" i "btn2") do wyboru konkretnej operacji oraz pole tekstowe, w którym wyświetlany będzie wynik operacji.

**Aplikacja powinna mieć następujące funkcjonalności:**

1. Po naciśnięciu przycisku "btn1", jeśli pole tekstowe nie jest puste, aplikacja powinna obliczać kolejne potęgi liczby z pola editText. Wynik powinien być wyświetlany w polu tekstowym.
2. Po naciśnięciu przycisku "btn2", jeśli pole tekstowe nie jest puste, aplikacja powinna obliczyć pierwiastek kwadratowy z wprowadzonej liczby. Wynik powinien być wyświetlany w polu tekstowym.
3. Po wprowadzeniu liczby do pola tekstowego ("editText") i zmianie wartości tego pola, zmienna "pot" powinna być zresetowana do 1.
4. Jeśli pole tekstowe jest puste, aplikacja powinna wyświetlić krótkie powiadomienie ("Toast") informujące użytkownika o konieczności wprowadzenia wartości.



```
edit.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) { }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) { pot=1; }

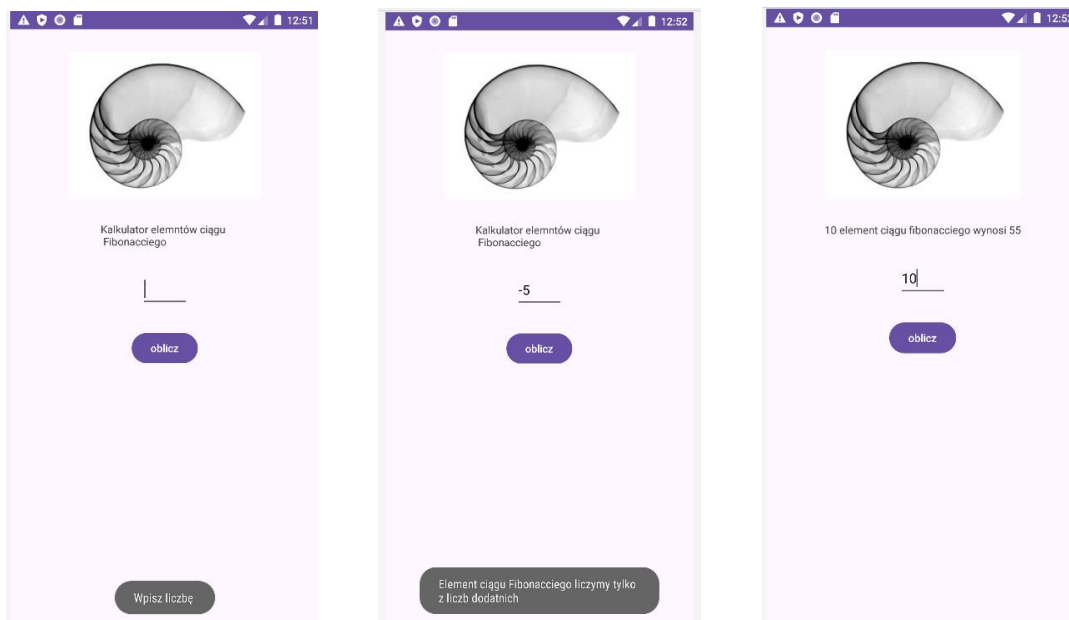
    @Override
    public void afterTextChanged(Editable s) { }
});
```

1 usage

```
private void potega( String editGet)
{
    int liczba = Integer.parseInt(editGet);
    String wynik = liczba + "^ " + pot + " = " + (int) Math.pow(liczba,pot);
    text.setText(wynik);
    pot++;
}
```

### Zadanie 3. Kalkulator Elementu w Ciągu Liczb Fibonacciego

Aplikacja obejmuje stworzenie interaktywnego interfejsu użytkownika dla kalkulatora elementu w ciągu liczb Fibonacciego, pozwalając na wprowadzenie liczby, a następnie obliczenie i wyświetlenie odpowiedniego elementu.



Twoim zadaniem jest stworzenie interaktywnego interfejsu użytkownika dla aplikacji mobilnej służącej do obliczania elementu w ciągu liczb Fibonacciego.

1. **Obrazek Fibonacci:** Umieść obraz reprezentujący ciąg liczb Fibonacciego ([@drawable/fibonacci](#)) na ekranie. To może być element wizualny identyfikujący cel aplikacji.
2. **Opis Aplikacji:** Wyświetl tekstowy opis aplikacji pod obrazkiem. Tekst ten powinien zawierać informację na temat funkcji kalkulatora, na przykład: "Kalkulator Elementu w Ciągu Liczb Fibonacciego".
3. **Pole Wprowadzania Liczby:** Dodaj pole tekstowe (**EditText**) umożliwiające wprowadzenie liczby, dla której chcesz obliczyć odpowiedni element ciągu Fibonacciego. Pole to powinno akceptować zarówno liczby dodatnie, jak i ujemne.
4. **Przycisk Obliczenia:** Dodaj przycisk (**Button**) oznaczony jako "Oblicz". Po naciśnięciu tego przycisku aplikacja powinna obliczyć i wyświetlić odpowiedni element ciągu Fibonacciego dla wprowadzonej liczby.
5. **Interfejs Responsywny:** Zapewnij, aby interfejs był responsywny i estetyczny, dostosowany do różnych rozmiarów ekranów.

## Dodatkowe Wskazówki:

- Upewnij się, że pole wprowadzania liczby akceptuje zarówno liczby całkowite dodatnie, jak i ujemne.
- W przypadku wprowadzenia nieprawidłowej liczby, wyświetl stosowne komunikaty błędów.
- Zaimplementuj logikę obliczeń elementu ciągu Fibonacciego po naciśnięciu przycisku "Oblicz".
- Przetestuj aplikację, aby sprawdzić, czy poprawnie obsługuje różne scenariusze.

3 usages

```
private int fibonacci(int n)
{
    if(n==0) return 0;
    else if(n==1) return 1;
    else return fibonacci(n-2)+fibonacci(n-1);
}
```

```
btn.setOnClickListener(v -> {
    String ele = edit.getText().toString().trim();

    if(!ele.isEmpty()) {
        int element = Integer.parseInt(ele);

        if (element < 0)
            Toast.makeText(context: this, text: "Element ciągu Fibonacciego liczymy tylko z liczb dodatnich", Toast.LENGTH_LONG).show();
        else {
            int fib = fibonacci(element);
            text.setText(element+"element ciągu fibonacciego wynosi "+fib);
            //text.setText(String.format("%d element ciągu fibonacciego wynosi %d", element, fib));
        }
    }else Toast.makeText(context: this, text: "Wpisz liczbę ", Toast.LENGTH_SHORT).show();
});
```

## **Zadanie 4. Kalkulator Kosztów Wakacyjnych.**

Aplikacja pobiera dane wejściowe od użytkownika, takie jak liczba osób, liczba dni, koszt zakwaterowania, transportu oraz wyżywienia. Na podstawie tych danych, obliczany jest całkowity koszt wyjazdu.

Dodatkowo, użytkownik może zaznaczyć pole wyboru, które zwiększa koszt o 20%, aby uwzględnić dodatkowe wydatki.

### **Szczegóły aplikacji:**

#### **1. Pola wejściowe:**

- ile\_osob: Liczba osób biorących udział w wyjeździe.
- ile\_dni: Liczba dni wyjazdu.
- zakwaterowanie: Koszt zakwaterowania za osobę na dzień.
- transport: Koszt transportu za osobę.
- wyżywienie: Koszt wyżywienia za osobę na dzień.
- dodatkowe: CheckBox zaznaczany, jeśli użytkownik chce uwzględnić dodatkowe wydatki, całkowity koszt zwiększa się o 20%.

#### **2. Przycisk obliczania:**

- Po kliknięciu tego przycisku, aplikacja oblicza całkowity koszt wyjazdu.

#### **3. Wynik:**

- Pole tekstowe, w którym wyświetlany jest wynik obliczeń.

### **Logika działania:**

#### **1. Po kliknięciu przycisku wywoływana jest metoda koszty(), która:**

- Pobiera wartości z pól wejściowych.
- Przelicza całkowity koszt wyjazdu jako sumę kosztów zakwaterowania, wyżywienia i transportu.
- Jeśli CheckBox dodatkowe jest zaznaczony, koszt jest zwiększany o 20%.

- Pole określające liczbę osób i liczbę dni nie może być puste i może akceptować tylko liczby całkowite

## 2. Metoda koszty():

- Pobiera wartości z pól tekstowych i konwertuje je na odpowiednie typy (int, double).
- Oblicza całkowity koszt jako  $(\text{osoby} * (\text{zakwaterowanie} + \text{wyżywienie}) * \text{dni}) + (\text{transport} * \text{osoby})$
- Opcjonalnie całkowity koszt jako  $(\text{osoby} * (\text{zakwaterowanie} + \text{wyżywienie}) * \text{dni}) + (\text{transport} * \text{osoby}) * 120\%$
- Zwraca obliczony koszt.

Rys 1.

4  
10  
120  
50  
80

dodatkowe koszty

Oblicz

Szacowany koszt wakacyjnego wyjazdu:  
8200.0 zł

1 2 3 -  
4 5 6 -  
7 8 9 x  
, 0 . ✓

Rys 2.

4  
10  
120  
50  
80

dodatkowe koszty

Oblicz

Szacowany koszt wakacyjnego wyjazdu:  
9840.0 zł

1 2 3 -  
4 5 6 -  
7 8 9 x  
, 0 . ✓

Rys 3.

2  
Czas trwania (w dniach)  
100  
Koszt transportu (na osobę)  
Koszt wyżywienia (na osobę za dzień)

dodatkowe koszty

Oblicz

Szacowany koszt wakacyjnego wyjazdu:  
wyjazdu: 0.0 zł

liczba osób lub dni nie spełnia wymogów