

Ćwiczenia 7,8 — Android studio – AnalogClock_TextClock_setOnClickListener_Thread_Alarm_Switch

Na koniec zajęć prześlij pliki źródłowe (*.xml, *.java)+ obrazek do zasobu w teams.

1. Utwórz nowy projekt o nazwie Alarm na podstawie Empty Activity (dobrać odpowiednie API).
2. Otworzyć dokumentację:

<https://developer.android.com/training/scheduling/alarms>

<https://developer.android.com/reference/android/app/AlarmManager>

<https://developer.android.com/reference/android/content/BroadcastReceiver>

<https://developer.android.com/reference/android/app/PendingIntent>

<https://developer.android.com/reference/android/os/Vibrator>

3. Pamiętaj, aby w pliku values\strings.xml i values\colors.xml umieścić kolory i teksty.
4. Dodaj komponenty:
 - a) TimePicker, do ustawienia czasu alarmu
 - b) SwitchCompat, do włączenia/ wyłączenia
 - c) TextView do kontrolnego wyświetlenia wybranego czasu budzika
5. Użyj klasy Toast dla wyświetlenia czy alarm został włączony/wyłączony.
6. Dodaj w onCreate() **alarmManager** = (AlarmManager) getSystemService(**ALARM_SERVICE**);
7. Dodaj dla switcha listener: setOnClickListener
8. Utwórz metodę switchClicked(), którą wywołasz w onClick(View v){}
9. Użyj klasy Calendar do ustawienia czasu.

```
// Set the alarm to start at approximately 2:00 p.m.  
Calendar calendar = Calendar.getInstance();  
calendar.setTimeInMillis(System.currentTimeMillis());  
calendar.set(Calendar.HOUR_OF_DAY, 14);
```

10. Dodaj nową klasę o nazwie MyBroadcastReceiver, która rozszerza klasę BroadcastReceiver, następnie zaimplementuj metodę

@Override

public void onReceive(Context context, Intent intent) {}

11. Utwórz obiekt klasy Vibrator w metodzie onReceive i przypisz getSystemService(context.**VIBRATOR_SERVICE**) dla context
12. Dla metody vibrate() dodaj

[https://developer.android.com/reference/android/os/VibrationEffect#createOneShot\(long,%20int\)](https://developer.android.com/reference/android/os/VibrationEffect#createOneShot(long,%20int))

13. Zaakceptuj dodanie `<uses-permission android:name="android.permission.VIBRATE" />`

14. Dla Uri dodaj fragment kodu:

```
AtomicReference<Uri> alarmUri = new AtomicReference<>(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM));
if (alarmUri.get() == null) {
    alarmUri.set(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION));
}
```

15. Utwórz obiekt klasy Ringtone, <https://developer.android.com/reference/android/media/Ringtone?hl=ja>
a dla RingtoneManagera

[https://developer.android.com/reference/android/media/RingtoneManager#getRingtone\(android.content.Context,%20android.net.Uri\)](https://developer.android.com/reference/android/media/RingtoneManager#getRingtone(android.content.Context,%20android.net.Uri))

16. Wywołaj play().

17. W metodzie switchClicked(){} stwórz Intent dla MyBroadcastReceiver jak poniżej:

Wake up the device to fire the alarm at precisely 8:30 a.m., and every 20 minutes thereafter:

Kotlin

Java

```
private AlarmManager alarmMgr;
private PendingIntent alarmIntent;
...
alarmMgr = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
Intent intent = new Intent(context, AlarmReceiver.class);
alarmIntent = PendingIntent.getBroadcast(context, 0, intent, 0);

// Set the alarm to start at 8:30 a.m.
Calendar calendar = Calendar.getInstance();
calendar.setTimeInMillis(System.currentTimeMillis());
calendar.set(Calendar.HOUR_OF_DAY, 8);
calendar.set(Calendar.MINUTE, 30);

// setRepeating() lets you specify a precise custom interval--in this case,
// 20 minutes.
alarmMgr.setRepeating(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
    1000 * 60 * 20, alarmIntent);
```

18. Dodaj do pliku AndroidManifest.xml, receiver:

<https://developer.android.com/guide/topics/manifest/receiver-element>

19. Dodaj powiadomienie.

20. Otwórz dokumentację:

<https://developer.android.com/training/notify-user/build-notification>

<https://developer.android.com/reference/android/app/NotificationChannel>

<https://developer.android.com/training/notify-user/channels>

<https://developer.android.com/training/notify-user/expanded>

<https://developer.android.com/training/notify-user/navigation>

21. Dodaj Empty Activity o nazwie Notification do realizacji wyskakującego powiadomienia.
22. Dodaj Intent dla Notification.class w **public void** onReceive(Context context, Intent intent) {}
23. Dla intentu ustaw flagi Intent.**FLAG_ACTIVITY_NEW_TASK** | Intent.**FLAG_ACTIVITY_CLEAR_TASK**
24. Utwórz PendingIntent z getActive() na Intent.
25. Ustaw treść powiadomienia w metodzie **public void** onReceive(Context context, Intent intent) {}

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle(textTitle)
    .setContentText(textContent)
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

26. Powinieneś uzyskać coś na kształt:

```
// Create an explicit intent for an Activity in your app
Intent intent = new Intent(this, AlertDetails.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);

NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle("My notification")
    .setContentText("Hello World!")
    .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    // Set the intent that will fire when the user taps the notification
    .setContentIntent(pendingIntent)
    .setAutoCancel(true);
```

27. Dodaj setStyle

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle("My notification")
    .setContentText("Much longer text that cannot fit one line...")
    .setStyle(new NotificationCompat.BigTextStyle()
        .bigText("Much longer text that cannot fit one line..."))
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

28. Dodaj kanał powiadomień w MainActivity.java w metodzie **private void** createNotificationChanel(){}

```
private void createNotificationChannel() {  
    // Create the NotificationChannel, but only on API 26+ because  
    // the NotificationChannel class is new and not in the support library  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        CharSequence name = getString(R.string.channel_name);  
        String description = getString(R.string.channel_description);  
        int importance = NotificationManager.IMPORTANCE_DEFAULT;  
        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);  
        channel.setDescription(description);  
        // Register the channel with the system; you can't change the importance  
        // or other notification behaviors after this  
        NotificationManager notificationManager = getSystemService(NotificationManager.class);  
        notificationManager.createNotificationChannel(channel);  
    }  
}
```

29. Wywołaj powyższą metodę zaraz po starcie aplikacji.(w onCreate())

30. Przetestuj działanie aplikacji.

31. Zmodyfikuj metodę dla alarmu (zamień setRepeating na inną, zmień też RTC_WAKEUP).

32. Zadania dodatkowe:

a) zaprojektuj uruchamianie alarmu nawet po wyłączeniu urządzenia

wskazówka. 1

```
//  
if (intent.getAction().equals("android.intent.action.BOOT_COMPLETED")) {  
    // Set the alarm here.  
}
```

wskazówka 2

<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>

b) dodaj obsługę wyłączenia alarmu

c) dodaj obsługę drzemki SNOOZE

```
Intent snoozeIntent = new Intent(this, MyBroadcastReceiver.class);
snoozeIntent.setAction(ACTION_SNOOZE);
snoozeIntent.putExtra(EXTRA_NOTIFICATION_ID, 0);
PendingIntent snoozePendingIntent =
    PendingIntent.getBroadcast(this, 0, snoozeIntent, 0);

NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle("My notification")
    .setContentText("Hello World!")
    .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    .setContentIntent(pendingIntent)
    .addAction(R.drawable.ic_snooze, getString(R.string.snooze),
        snoozePendingIntent);
```

33. KONIEC.