**[알람 기능 구현하기]**

1. **알람매니저 생성**

AlarmManager **alarmManager =** (AlarmManager) getSystemService(***ALARM\_SERVICE***);

알람을 받을 BroadcastReceiver인 AlarmReceiver를 호출할 인텐트를 생성한다.

Intent **intent** = **new** Intent(**this**,AlarmReceiver.**class**);

1. **알람 시간 설정**

캘린더 객체를 이용하여 시간을 설정한다.

Calendar **calendar** = Calendar.*getInstance*();  
**calendar**.set(Calendar.***HOUR\_OF\_DAY***,13);  
**calendar**.set(Calendar.***MINUTE***,time);  
**calendar**.set(Calendar.***SECOND***,0);

\*현재 시간보다 설정 시간이 과거인 경우 알람이 무조건 울리게 된다. 따라서 현재 시간에서 설정한 시간을 빼서 설정시간이 과거인지 미래인지를 체크한다.

**long** diff = Calendar.*getInstance*().getTimeInMillis()-**calendar**.getTimeInMillis();

**if**(diff>0){  
 **calendar**.add(Calendar.***DAY\_OF\_WEEK***,1); //설정할 시간이 과거일 경우 설정 시간에 하루를 더함  
}

1. **시스템에 알람 설정**

알람이나 notificatrion에 Intent를 등록하기 위해 PendingIntent를 설정한다. 이 때 두번째 파라미터인 requestCode는 알람을 구분할 수 있는 id로 활용할 수 있다. 따라서 여러 개의 알람을 동시에 설정할 때 유니크한 requestCode값을 주면 알람이 구분되어 설정된다. 만약 같은 requestCode를 가진 PendingIntent로 알람을 여러 번 설정하면 계속 업데이트되어 마지막 알람만 설정된다. 설정된 알람을 해제할 때도 이 requestCode를 이용하여 특정 알람을 해제할 수 있다.

마지막에 PendingIntent.***FLAG\_UPDATE\_CURRENT*** 는 해당 PendingIntent가 있으면 업데이트를 시키라는 뜻이다.

PendingIntent **pendingIntent**

= PendingIntent.*getBroadcast*(**this**,9,**intent**,PendingIntent.***FLAG\_UPDATE\_CURRENT***);

알람매니저에 시간 기준, 동작할 시간, PendingIntent를 파라미터로 주어 알람을 시스템에 설정한다. Alarm은 어플리케이션과 생명주기를 같이 하는 것이 아닌 OS와 생명주기를 같이 한다.

**alarmManager**.setExact(AlarmManager.***RTC***,**calendar**.getTimeInMillis(),**pendingIntent**);

\*이미 설정된 알람이 있는지 확인하는 방법

*//각각의 requestCode로 설정된 PendingIntent를 확인하여 이미 알람 셋팅이 되어있는지 확인->하나라도 안돼있으면 알람 설정하는 메소드호출***if**(PendingIntent.*getBroadcast*(**this**, 9, **intent**, PendingIntent.***FLAG\_NO\_CREATE***)==**null**||  
 PendingIntent.*getBroadcast*(**this**, 14, **intent**, PendingIntent.***FLAG\_NO\_CREATE***)==**null**||  
 PendingIntent.*getBroadcast*(**this**, 18, **intent**, PendingIntent.***FLAG\_NO\_CREATE***)==**null**){  
 setTimer();//알람 설정메소드  
 }

1. **AlarmRecevier 구현**

알람이 울리면 설정한 Intent의 BroadcastReceiver가 호출된다. AlarmReceiver의 onReceive 안에 알람이 울리면 실행할 동작을 설정한다. 단 BroadcastReceiver에서는 10초 후 timout되므로 10초 이내의 짧은 작업을 처리하는 코드만 구현한다. 따라서 서비스를 호출하는 방식으로 무거운 작업을 처리한다..

**public class** AlarmReceiver **extends** BroadcastReceiver {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {

}

}