|  |
| --- |
| **학번 : 201911204, 이름 : 이주형** |
| **2023년 12주차 실습 체크 리스트** |
| |  |  | | --- | --- | | **실습 1** | **완성** | | **실습 2** | **완성** | |
| **실습 1 : [ 정규식 표현 ]** |
| **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: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="Hello World!"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |
| **MainActivity.kt** |
| package com.frost.lab01regularexpression  import android.content.BroadcastReceiver import android.content.Context import android.content.Intent import android.content.IntentFilter import android.content.pm.PackageManager import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import android.provider.Settings import android.widget.Toast import androidx.activity.result.contract.ActivityResultContracts import androidx.appcompat.app.AlertDialog import androidx.core.app.ActivityCompat import com.frost.lab01regularexpression.databinding.ActivityMainBinding  class MainActivity : AppCompatActivity() {  lateinit var binding: ActivityMainBinding  lateinit var broadcastReceiver: BroadcastReceiver  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivityMainBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)  initPermission()  initLayout()  //getMessage(intent)  checkSettingOverlayWindow(*intent*)  //activity가 처음부터 create될 때 실행되는 경우  }   override fun onNewIntent(intent: Intent?) {  //activity가 이미 생성되어 있고 실행중인 경우  super.onNewIntent(intent)  checkSettingOverlayWindow(intent)  }  fun getMessage(intent: Intent?){  if(intent!=null){  if(intent.hasExtra("msgSender") and intent.hasExtra("msgBody")){//확인  val msgSender = intent.getStringExtra("msgSender")  val msgBody = intent.getStringExtra("msgBody")  Toast.makeText(this, "보낸 번호 : "+msgSender+"\n"+ msgBody, Toast.*LENGTH\_SHORT*).show()  }  }  }  fun checkSettingOverlayWindow(intent: Intent?){  if(Settings.canDrawOverlays(this)){  getMessage(intent)  }else{  val overlayIntent = Intent(Settings.*ACTION\_MANAGE\_OVERLAY\_PERMISSION*)//화면 전환  requestSettingLauncher.launch(overlayIntent)  }  }   val requestSettingLauncher =  registerForActivityResult(ActivityResultContracts.StartActivityForResult())**{** if(Settings.canDrawOverlays(this)){  getMessage(this.*intent*)  }else{  Toast.makeText(this, "권한 승인 거부되었습니다.", Toast.*LENGTH\_SHORT*).show()  }  **}** val requestPermissionLauncher =  registerForActivityResult(ActivityResultContracts.RequestPermission())**{** if(**it**){  initPermission()  }else{  Toast.makeText(this, "권한 승인 거부되었습니다.", Toast.*LENGTH\_SHORT*).show()  }  **}** private fun permissionAlertDlg() {  val builder = AlertDialog.Builder(this)  builder.setMessage("반드시 문자 수신 권한이 허용되어야 합니다.")  .setTitle("권한 체크")  .setPositiveButton("OK")**{**//허용 버튼 누를 시  \_, \_ **->** requestPermissionLauncher.launch(android.Manifest.permission.*RECEIVE\_SMS*)  **}**.setNegativeButton("Cancel")**{** dlg, \_ **->** dlg.dismiss()//다이얼로그 종료  **}** builder.create().show()//builder 객체인 다이얼로그를 만들고 출력  }   private fun initPermission() {  when{  (ActivityCompat.checkSelfPermission(this,  android.Manifest.permission.*RECEIVE\_SMS*) == PackageManager.*PERMISSION\_GRANTED*)->{  Toast.makeText(this, "문자 수신 동의함", Toast.*LENGTH\_SHORT*).show()  }  ActivityCompat.shouldShowRequestPermissionRationale(this,  android.Manifest.permission.*RECEIVE\_SMS*)->{  permissionAlertDlg()  }else -> {  requestPermissionLauncher.launch(android.Manifest.permission.*RECEIVE\_SMS*)  }  }  }   private fun initLayout() {  broadcastReceiver = object: BroadcastReceiver(){  override fun onReceive(context: Context?, intent: Intent?) {  if(intent!=null){  if(intent.*action*.*equals*(Intent.*ACTION\_POWER\_CONNECTED*)){  Toast.makeText(context, "충전 시작", Toast.*LENGTH\_SHORT*).show()  }else if(intent.*action*.*equals*(Intent.*ACTION\_POWER\_DISCONNECTED*)){  Toast.makeText(context, "충전 해제", Toast.*LENGTH\_SHORT*).show()  }  }  }  }  }   override fun onResume() {  //메세지 수신  super.onResume()  val intentFilter = IntentFilter(Intent.*ACTION\_POWER\_CONNECTED*)  //충전기에 꽂을 때의 정보 수신  intentFilter.addAction(Intent.*ACTION\_POWER\_DISCONNECTED*)  //충전기에 뺐을 때 정보 수신  registerReceiver(broadcastReceiver, intentFilter)  }   override fun onPause() {  //메세지 수신 해제  super.onPause()  unregisterReceiver(broadcastReceiver)  } } |
| **MyReceiver.kt** |
| package com.frost.lab01regularexpression  import android.content.BroadcastReceiver import android.content.Context import android.content.Intent import android.provider.Telephony import android.util.Log import kotlinx.coroutines.CoroutineScope import kotlinx.coroutines.Dispatchers import kotlinx.coroutines.launch  class MyReceiver : BroadcastReceiver() {  //val pattern1 = Regex("^\\d{2}")  //^로 시작함을 의미. d는 숫자 {2}는 2자리 의미  val pattern1 = Regex("""^\d{2}/\d{2}\s\d{2}:\d{2}\s+.+원$""")//이런식으로 표현도 가능 """~"""  //정규식 객체로 패턴  val scope = *CoroutineScope*(Dispatchers.IO)   override fun onReceive(context: Context, intent: Intent) {  val pendingResult = goAsync()  //이 함수가 finish되기 전까지는 시스템에 의해서 강제 종료되지 않음  scope.launch **{** //문자가 오면 해당 메서드가 호출됨 이때 intent에 관련된 정보를 Callback  if (intent.*action*.*equals*("android.provider.Telephony.SMS\_RECEIVED")) {  val msg = Telephony.Sms.Intents.getMessagesFromIntent(intent)  val message = msg[0].*messageBody* //해당 메세지에 내가 원하는 문자열 있는지 비교하면 된다.   if (message.*contains*("건국카드")) {  val tempstr = message.*split*("\n")  //줄단위로 split  //for(str in tempstr.subList(1, tempstr.size)){  //일부 줄만 갖고 올 수도 있음  var result = false  for (str in tempstr) {  if (pattern1.containsMatchIn(str)) {//패턴 비교  result = true  break  }  }  if (result) {  val newIntent = Intent(context, MainActivity::class.*java*)  newIntent.*flags* =  Intent.*FLAG\_ACTIVITY\_NEW\_TASK* or Intent.*FLAG\_ACTIVITY\_SINGLE\_TOP* or Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* //launch mode 지정  newIntent.putExtra("msgSender", msg[0].*originatingAddress*)  newIntent.putExtra("msgBody", msg[0].*messageBody*)  context.startActivity(newIntent)  }  }  }  **}** pendingResult.finish()  //코루틴 스코프가 끝났으니 종료 보장  } } |
| **실행결과** |
|  |