

SE310 – Утасгүй хөдөлгөөнт програмчлал

Багш Ж.Золжаргал



Диалогын тухай

- Дэсктоп болон вэб Аппликешн-д байдагтай төстэй
- Асуулт асуух, сонголт хийх, үйлдэл баталгаажуулах, анхааруулга болон алдааны мэссэж уншуулах байдлаар ашигладаг



Диалогын тухай



Dialog классын тухай



- Dialog класс нь энгийн цонх хэлбэрээр харагддаг бөгөөд Activity класс-д харагдана

```
Dialog d = new Dialog(MyActivity.this);
```

```
// Have the new window tint and blur the window it  
// obscures.  
Window window = d.getWindow();  
window.setFlags(WindowManager.LayoutParams.FLAG_BLUR_BEHIND,  
                WindowManager.LayoutParams.FLAG_BLUR_BEHIND);
```

```
d.setTitle("Dialog Title");  
d setContentView(R.layout.dialog_view);
```

```
TextView text = (TextView)d.findViewById(R.id.dialogTextView);  
text.setText("This is the text in my dialog");
```

```
d.show();
```

AlertDialog класс



- Энэхүү класс нь хувирамтгай бөгөөд олон янзаар ашиглах боломжтой
 - Дэлгэцэнд мэссэжийг 1-3 сонгох товчлууртай харуулдаг
 - Check button болон Radio button-уудын сонголтуудыг харуулдаг
 - Текст оруулах Input контролоор хангадаг
- Alert Dialog-н user interface цогцлоох

```
AlertDialog.Builder ad = new AlertDialog.Builder(context);
```

```
Context context = MyActivity.this;
String title = "It is Pitch Black";
String message = "You are likely to be eaten by a grue.";
String button1String = "Go Back";
String button2String = "Move Forward";

AlertDialog.Builder ad = new AlertDialog.Builder(context);
ad.setTitle(title);
ad.setMessage(message);
ad.setPositiveButton(button1String,
    new OnClickListener() {
        public void onClick(DialogInterface dialog,
                            int arg1) {
            eatenByGrue();
        }
    });
ad.setNegativeButton(button2String,
    new OnClickListener(){
        public void onClick(DialogInterface dialog,
                            int arg1) {
            // do nothing
        }
    });
ad.setCancelable(true);
ad.setOnCancelListener(new OnCancelListener() {
```



AlertDialog класс



```
public void onCancel(DialogInterface dialog) {  
    eatenByGrue();  
}  
});
```

- Дэлгэцэнд үзүүлэх

```
ad.show();
```

- Хэрэглэгч *onCreateDialog* болон *onPrepareDialog* функцуудыг өөрийн Activity-д даран тодорхойлох боломжтой

Input Dialogs



- Диалогын чулах хэрэглээний нэг бол хэрэглэгчээс өгөгдөл авах
- **DatePickerDialog** – DatePicker контролоос огноо сонгох
- **TimePickerDialog** – TimePicker контролоос цаг сонгох
- **ProgressDialog** – Одоо явагдаж буй процессийн явцыг харуулахад тохиромжтой

Dialog-уудыг хэрэглэх болон удирдах

- Класс-д onCreateDialog функцыг даран тодорхойлсоноор тухайн Dialog-г дэлгэцэнд харуулахад showDialog функц хэрэглэгдэнэ

```
static final private int TIME_DIALOG = 1;

@Override
public Dialog onCreateDialog(int id) {
    switch(id) {
        case (TIME_DIALOG) :
            AlertDialog.Builder timeDialog = new AlertDialog.Builder(this);
            timeDialog.setTitle("The Current Time Is...");
            timeDialog.setMessage("Now");
            return timeDialog.create();
        }
    return null;
}
```

Dialog-уудыг хэрэглэх болон удирдах

```
@Override
public void onPrepareDialog(int id, Dialog dialog) {
    switch(id) {
        case (TIME_DIALOG) :
            SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
            Date currentTime;
            currentTime = new Date(java.lang.System.currentTimeMillis());
            String dateString = sdf.format(currentTime);
            AlertDialog timeDialog = (AlertDialog)dialog;
            timeDialog.setMessage(dateString);

            break;
    }
}
```

- Dialog-ийг дуудах

```
showDialog(TIME_DIALOG);
```

Activity-г диалог байдлаар ашиглах

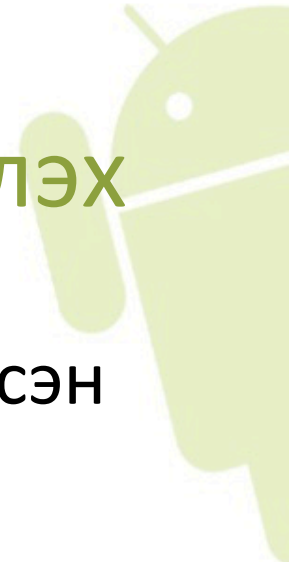


- Dialog box-н контролууд болон амьдралын муж нь хангалтгүй үед
- Activity класс-д байдаг бүхий л боломж байна. Амьдалын мужын бүхий л Event-үүд
- Activity классыг Диалог шиг болгоход Manifest XML файлд

```
<activity android:name="MyDialogActivity"  
          android:theme="@android:style/Theme.Dialog">  
</activity>
```

Intent-г Broadcast Event-тэй хэрэглэх

- Системийн түвшинд мэссэж дамжуулна гэсэн үг
- Бүтэцлэгдсэн мэссэжийг процессуудруу илгээнэ
- `sendBroadcast` функцын тусламжтай бүрэлдхүүн хэсгүүдэд broadcast мэссэж илгээдэг
- Broadcast Receiver-г хэрэгжүүлэх шаардлагатай бөгөөд чагнах болон хариу үйлдэл үзүүлэх боломжтой



Broadcast event

- Ирсэн дуудлага чагнах
- Утасны хонх солих
- Батарей хэмжээ
- Сүлжээн холбогдсон эсэх



Broadcast Intent



- Broadcast Intent нь энгийн бөгөөд sendBroadcast функцыг ашиглана
- Action string нь жава пакежын нэрийг агуулна

```
public static final String NEW_LIFEFORM_DETECTED =  
    "com.paad.action.NEW_LIFEFORM";
```

```
Intent intent = new Intent(NEW_LIFEFORM_DETECTED);  
intent.putExtra("lifeformName", lifeformType);  
intent.putExtra("longitude", currentLongitude);  
intent.putExtra("latitude", currentLatitude);
```

```
sendBroadcast(intent);
```

Listening for Broadcasts with Broadcast Receivers



- Broadcast Receiver-г идэвхжүүлэхийн тулд manifest XML файлд бүртгүүлэх шаардлагатай

```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class MyBroadcastReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        //TODO: React to the Intent received.
    }

}
```

Listening for Broadcasts with Broadcast Receivers



- Broadcast Intent ирэхэд onReceive функц ажиллах бөгөөд 5 секундын дотор ажиллаж дуусах ёстой
- Дуусаагүй тохиолдол *Application Unresponsive* диалог гарч ирнэ
- Broadcast-аар тухайн Intent орж ирэхэд BroadcastReceiver класс автоматаар ажиллаж эхэлдэг

Listening for Broadcasts with Broadcast Receivers



- Энэ нь Event-ээр удирдагдсан апп бичих боломжийг олгодог бөгөөд хаагдсан эсвэл устсан байсан ч ажиллах чадвартай
- Ингэснээр санах ойн нөөцийн маш сайн ашиглалт болдог
- Ерөнхийдөө Broadcast Receiver-г мэдээллийг шинэчлэх, Service-г эхлүүлэх, Activity UI шинэчлэх болон хэрэглэгчид мэдэгдэл хүргэх байдлаар ашигладаг.

Жишээ

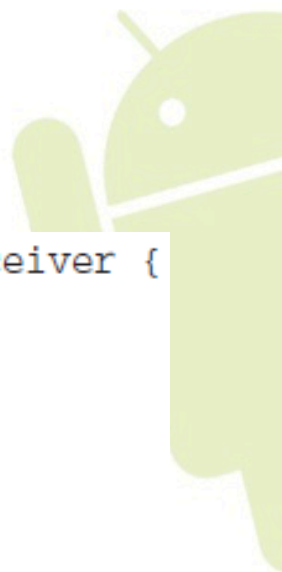
```
public class LifeformDetectedBroadcastReceiver extends BroadcastReceiver {

    public static final String BURN =
        "com.paad.alien.action.BURN_IT_WITH_FIRE";

    @Override
    public void onReceive(Context context, Intent intent) {
        // Get the lifeform details from the intent.
        Uri data = intent.getData();
        String type = intent.getStringExtra("type");
        double lat = intent.getDoubleExtra("latitude", 0);
        double lng = intent.getDoubleExtra("longitude", 0);
        Location loc = new Location("gps");
        loc.setLatitude(lat);
        loc.setLongitude(lng);

        if (type.equals("alien")) {
            Intent startIntent = new Intent(BURN, data);
            startIntent.putExtra("latitude", lat);
            startIntent.putExtra("longitude", lng);

            context.startActivity(startIntent);
        }
    }
}
```



Manifest XML-д бүртгүүлэх



- Broadcast Receiver-г бүртгүүлэхэд Application tag-д Receiver tag байрлана.
- Receiver tag нь intent-filter tag агуулах бөгөөд үүнд Action string-г зааж өгнө

```
<receiver android:name=".LifeformDetectedBroadcastReceiver">  
  <intent-filter>  
    <action android:name="com.paad.action.NEW_LIFEFORM" />  
  </intent-filter>  
</receiver>
```

Broadcast Receiver-г жава кодноос бүртгүүлэх



```
// Create and register the broadcast receiver.  
IntentFilter filter = new IntentFilter(NEW_LIFEFORM_DETECTED);  
LifeformDetectedBroadcastReceiver r = new LifeformDetectedBroadcastReceiver()  
registerReceiver(r, filter);
```

- Бүртгэлийг цуцлах

```
unregisterReceiver(r);
```

Андройдын Broadcast Action-ууд

- ACTION_BOOT_COMPLETED – Систем ачааллаж дуусахад нэг л удаа үүснэ. RECEIVE_BOOT_COMPLETED зөвшөөрөл шаардлагатай
- ACTION_CAMERA_BUTTON – Камерын товчлуур дарагдахад
- ACTION_DATE_CHANGED,
ACTION_TIME_CHANGED



Андройдын Broadcast Action-ууд

- ACTION_GTALK_SERVICE_CONNECTED, ACTION_GTALK_SERVICE_DISCONNECTED – Gtalk-н холболт салах болон нийлэх бүрт үүснэ
- ACTION_MEDIA_BUTTON – Медиа товчлуур дарагдахад
- ACTION_MEDIA_EJECT – Гадаад хадгалах төхөөрөмжнөөс сонгоод гарах үед



Андройдын Broadcast Action-ууд



- ACTION_MEDIA_MOUNTED, ACTION_MEDIA_UNMOUNTED – Хадгалах төхөөрөмж нэмэгдэх болон хасагдах үед
- ACTION_SCREEN_OFF, ACTION_SCREEN_ON – Дэлгэц унтрах болон асах үед
- ACTION_TIMEZONE_CHANGED – Одоогын цагийн бүс өөрчлөгдөхөд
- <http://code.google.com/android/reference/android/content/Intent.html>

Internet-д холбогдох



- Тухайн апп интернетэд холбогдохын тул доорх зөвшөөрлийг авах шаардлагатай

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

- Жишээ нь

```
String myFeed = getString(R.string.my_feed);  
try {  
    URL url = new URL(myFeed);  
  
    URLConnection connection = url.openConnection();  
    HttpURLConnection httpConnection = (HttpURLConnection)connection;  
  
    int responseCode = httpConnection.getResponseCode();  
    if (responseCode == HttpURLConnection.HTTP_OK) {  
        InputStream in = httpConnection.getInputStream();
```


Internet-д холбогдох

```
        [ ... Process the input stream as required ... ]  
    }  
}  
catch (MalformedURLException e) { }  
catch (IOException e) { }
```

- Сүлжээний харилцаа холбоо үүсгэхэд дараах сангууд шаардлагатай
 - java.net.*
 - android.net.*



Интернет ашигласанаар



- Өөрийн Activity-д WebKit-based броузер буюу WebView контрол ашиглах боломжтой
- Клиентээс Сервер процессуудтай холбогдох Google's GData API-г ашиглах боломжтой
- Мөн XML-ээр мэдээлэл солилцоход туслах SAX болон javaх гэсэн Java-based XML хөрвүүлэгчид бий
- EDGE ба GSM болон Wi-Fi холболт

Програмын энгийн өгөгдөл хадгалах

- name/value pair (NVP) механизм
- *SharedPreferences* классыг ашиглах бөгөөд key/value гэсэн хослолоор тухайн програмын өгөгдлийг хадгалж програмын аль ч бүрэлдхүүн хэсэгт дуудаж ашиглах боломжтой
- Shared Preference нь өгөгдлийн үндсэн төрлүүдийг дэмждэг
 - Boolean, string, float, long, integer, классын тохиолдол, одоогын UI-н төлөв



Програмын энгийн өгөгдөл хадгалах

- Түгээмэл хэрэглээ нь
 - User session / хэрэглэгчийн мэдээлэл хадгалах
 - Share settings / програмын тохиргоо хадгалах
- Тухайн Activity-н төлөвийг *onSaveInstanceState* үзэгдлийн тусламжтай хадгалж авах боломжтой



Creating and Saving Preferences



- Shared Preference-г шинэчлэх болон үүсгэхдээ *getSharedPreferences* функцыг ашигладаг
- Shared Preference-д хадгаласан өгөгдөл рүү програмын аль ч бүрэлдхүүн хэсэгээс хандах боломжтой, гэвч өөр програмаас хандах боломжгүй
- Shared Preference-г шинэчлэхдээ *SharedPreferences.Editor* классыг ашигладаг

Creating and Saving Preferences

```
public static final String MYPREFS = "mySharedPreferences";

protected void savePreferences(){
    // Create or retrieve the shared preference object.
    int mode = Activity.MODE_PRIVATE;
    SharedPreferences mySharedPreferences = getSharedPreferences(MYPREFS,
                                                                mode);

    // Retrieve an editor to modify the shared preferences.
    SharedPreferences.Editor editor = mySharedPreferences.edit();

    // Store new primitive types in the shared preferences object.
    editor.putBoolean("isTrue", true);
    editor.putFloat("lastFloat", 1f);
    editor.putInt("wholeNumber", 2);
    editor.putLong("aNumber", 3l);
    editor.putString("textEntryValue", "Not Empty");

    // Commit the changes.
    editor.commit();
}
```

Retrieving Shared Preferences



- Хадгаласан өгөгдөл рүү хандахад мөн *getSharedPreferences* функцыг ашиглана
- Shared Preference-с утга авахдаа *key* дамжуулах бөгөөд *get<type>* гэсэн функцыг ашиглана
- Get функцууд нь 2 параметр авах бөгөөд *key* болон *default value* байна. (Тухайн *key* дээр ямар нэгэн утга байхгүй үед ашиглагдана)

}

Saving the Activity State



- Activity-н мэдээллийг хадгалахдаа програмын бусад бүрэлдхүүн хэсгүүдэд хуваалцахгүй тохиолдолд *Activity.getPreferences()* функцыг ашиглаж болно
- Activity бүрт нэг ширхэг нэргүй Shared Preference –н объект үүсгэх боломжтой

Saving the Activity State



```
protected void saveActivityPreferences() {  
    // Create or retrieve the activity preferences object.  
    SharedPreferences activityPreferences =  
        getPreferences(Activity.MODE_PRIVATE);  
  
    // Retrieve an editor to modify the shared preferences.  
    SharedPreferences.Editor editor = activityPreferences.edit();  
  
    // Retrieve the View  
    TextView myTextView = (TextView)findViewById(R.id.myTextView);  
  
    // Store new primitive types in the shared preferences object.  
    editor.putString("currentTextValue",  
        myTextView.getText().toString());  
  
    // Commit changes.  
    editor.commit();  
}
```

Saving and Restoring Instance State



- Shared Preference-тэй төстэй байдлаар Activity-н хувьсагчдыг хадгалж болно
- onSaveInstanceState функцыг даран тодорхойлох бөгөөд Bundle параметрийн тусламжтай өгөгдлийг хадгална
- Өгөгдөлд хандахдаа Shared Preference-тэй адил *get* болон *put* функцуудыг ашиглана

Saving and Restoring Instance State

```
private static final String TEXTVIEW_STATE_KEY = "TEXTVIEW_STATE_KEY";  
  
@Override  
public void onSaveInstanceState(Bundle outState) {  
    // Retrieve the View  
    TextView myTextView = (TextView) findViewById(R.id.myTextView);  
  
    // Save its state  
    outState.putString(TEXTVIEW_STATE_KEY,  
                       myTextView.getText().toString());  
    super.onSaveInstanceState(outState);  
}
```



Saving and Restoring Instance State



```
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.main);

    TextView myTextView = (TextView)findViewById(R.id.myTextView);

    String text = "";
    if (icle != null && icle.containsKey(TEXTVIEW_STATE_KEY))
        text = icle.getString(TEXTVIEW_STATE_KEY);

    myTextView.setText(text);
}
```