

# Android-Apps

## weitere Möglichkeiten

# weitere Möglichkeiten

I

N

F

O

R

M

A

T

I

K

- Speichern in der Sandbox
- onTouchListener
- Senden und Empfangen von SMS
- Anbindung von Datenbanken
- Der Beschleunigungssensor
- ...

- Übung

# Speichern in der Sandbox

I

N

F

O

R

M

A

T

I

K

- Externe Speicherkarte
  - Inhalt bleibt z.B. bei Rückstellung auf Werkeinstellung erhalten
  - Inhalt ist allgemein zugänglich (nicht geheim / veränderbar)
- Interner Speicher / Sandbox
  - Dateizugriffsrecht kann auf *MODE\_PRIVATE* gesetzt werden
  - Daten gehen bei Reset verloren
  - Speichermedium ist immer vorhanden
  - Speicherplatz ist begrenzt

# Speichern in der Sandbox

```
/* Überschreiben, um letzten Raum zu sichern */
```

```
@Override
```

```
protected void onStop() {
```

```
    FileOutputStream fos = null;
```

```
    OutputStreamWriter osw = null;
```

```
    try {
```

```
        fos = openFileOutput("aktudata.txt", MODE_PRIVATE);
```

```
        osw = new OutputStreamWriter(fos);
```

```
        osw.write(rbez);
```

```
    }
```

```
    catch (Throwable t) {
```

```
        Log.e("log_tag", "save "+t);
```

```
    }
```

```
    finally {
```

```
        if (osw != null) {
```

```
            try {
```

```
                osw.close();
```

```
            }
```

```
            catch (IOException e) {
```

```
                Log.e("log_tag", "osw.close "+e);
```

```
            }
```

```
        }
```

```
        if (fos != null) {
```

```
            try {
```

```
                fos.close();
```

```
            }
```

```
            catch (IOException e) {
```

```
                Log.e("log_tag", "fos.close "+e);
```

```
            }
```

```
        }
```

```
    }
```

```
    super.onStop();
```

```
}
```

Kein Verzeichnis?

→ Sandbox /

Verzeichnis der App

Nicht von  
anderen Apps  
zu lesen!

```
private void raumladen() {
```

```
    FileInputStream fis = null;
```

```
    InputStreamReader isr = null;
```

```
    BufferedReader br = null;
```

```
    try {
```

```
        fis = openFileInput("aktudata.txt");
```

```
        isr = new InputStreamReader(fis);
```

```
        br = new BufferedReader(isr);
```

```
        rbez = br.readLine();
```

```
    }
```

```
    catch (Throwable t) {
```

```
        Log.e("log_tag", "load "+t);
```

```
    }
```

```
    finally {
```

```
        if (br != null) {
```

```
            try {
```

```
                br.close();
```

```
            }
```

```
            catch (IOException e) {
```

```
                Log.e("log_tag", "br.close "+e);
```

```
            }
```

```
        }
```

```
        if (isr != null) {
```

```
            try {
```

```
                isr.close();
```

```
            }
```

```
            catch (IOException e) {
```

```
                Log.e("log_tag", "isr.close "+e);
```

```
            }
```

```
        }
```

```
        if (fis != null) {
```

```
            try {
```

```
                fis.close();
```

```
            }
```

```
            catch (IOException e) {
```

```
                Log.e("log_tag", "fis.close "+e);
```

```
            }
```

```
        }
```

# Speichern in der Sandbox

I

N

F

O

R

M

A

T

I

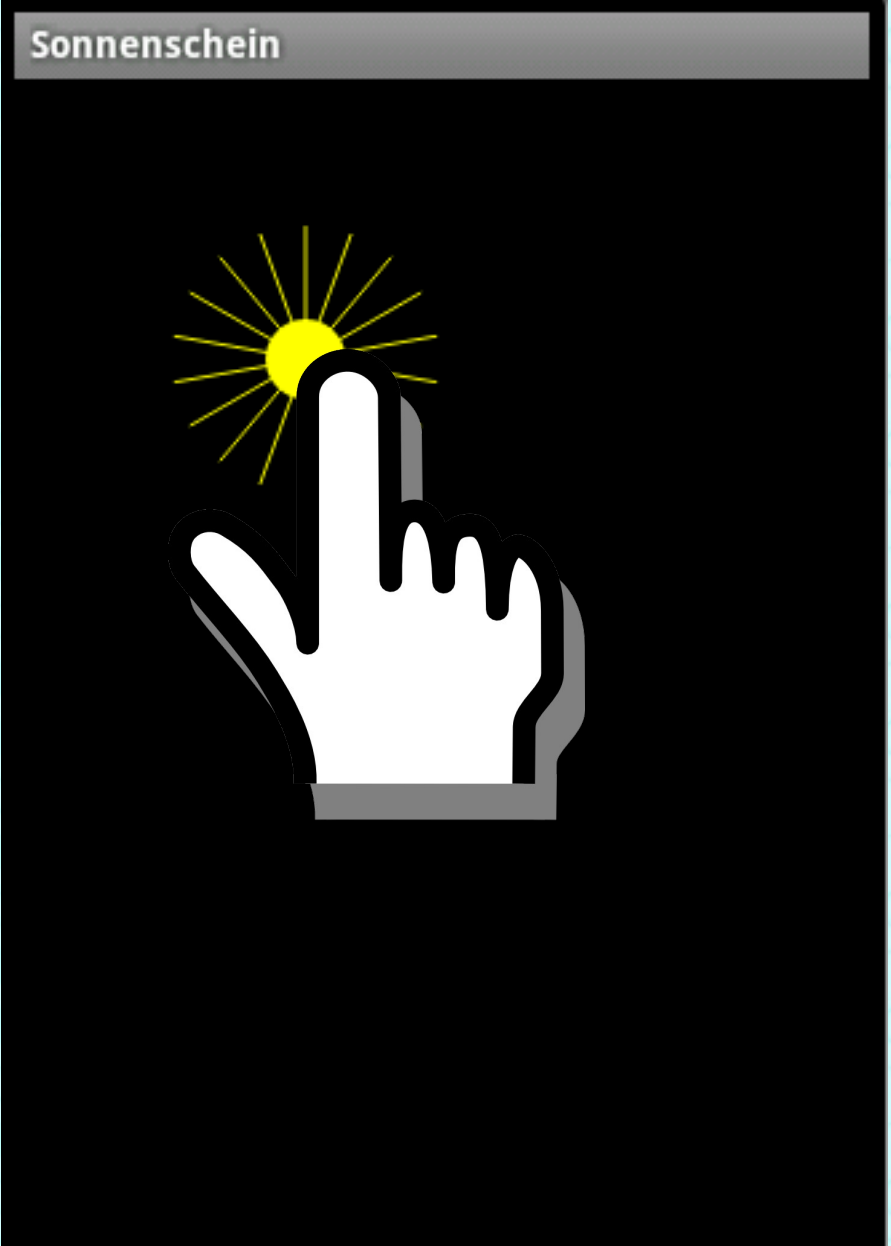
K

Name	Size	Date	Time	Permissions	Info
data		2011-05-15	11:05	drwxrwx--x	
app		2012-01-20	17:38	drwxrwx--x	
app-private		2011-05-15	11:04	drwxrwx--x	
backup		2011-05-15	11:05	drwx-----	
dalvik-cache		2012-01-20	17:38	drwxrwx--x	
data		2012-01-05	14:02	drwxrwx--x	
android.tts		2011-05-15	11:05	drwxr-x--x	
com.android.alarmclock		2011-05-15	11:06	drwxr-x--x	
com.android.browser		2012-01-02	13:13	drwxr-x--x	
...					
eh.beispiele.db		2012-01-12	14:32	drwxr-x--x	
files		2012-01-13	19:41	drwxrwx--x	
aktudata.txt	4	2012-01-18	11:53	<u>-rw-rw----</u>	
secdata.txt	16	2012-01-12	14:54	-rw-rw----	
lib		2012-01-05	14:02	drwxr-xr-x	
eh.beispiele.mathe		2012-01-02	09:56	drwxr-x--x	

Datei-  
verwaltung  
von Linux  
„geerbt“

# OnTouchListener

I  
N  
F  
O  
R  
M  
A  
T  
I  
K



# OnTouchListener

```
public class SonnenscheinDraw extends View implements OnTouchListener{
```

```
    Bitmap bitmap;
```

```
    Canvas bitmapCanvas;
```

eigene View definieren

```
    boolean isInitialized;
```

```
    // paint ist eher eine Art Stift
```

```
    Paint paint = new Paint();
```

```
    public SonnenscheinDraw(Context context){
```

```
        super(context);
```

```
        setFocusable(true);
```

```
        setFocusableInTouchMode(true);
```

wählbar machen

```
        this.setOnTouchListener(this);
```

```
        paint.setColor(Color.YELLOW);
```

```
        paint.setAntiAlias(true);
```

```
        paint.setStyle(Style.FILL_AND_STROKE);
```

```
        isInitialized = false;
```

```
    }
```

```
    private void init() {
```

```
        // RGB_565: 2 Bytes (16 Bit) für Rot(5), Grün(6) und Blau(5)
```

```
        bitmap = Bitmap.createBitmap(getWidth(), getHeight(), Bitmap.Config.RGB_565);
```

```
        bitmapCanvas = new Canvas();
```

```
        bitmapCanvas.setBitmap(bitmap);
```

```
        bitmapCanvas.drawColor(Color.BLACK);
```

```
        isInitialized = true;
```

```
    }
```

# OnTouchListener

```

@Override
public void onDraw(Canvas canvas) {
    // vor dem ersten Mal Initialisieren
    if (!isInitialized)
        init();

    canvas.drawBitmap(bitmap, 0, 0, paint);
}

public boolean onTouch(View view, MotionEvent event) {
    // weglassen (sowie Schleife), Kreis kleiner ... dann hat man ein Zeichenbrett
    bitmapCanvas.drawColor(Color.BLACK);
    float x = event.getX();
    float y = event.getY();
    float x2, y2;
    bitmapCanvas.drawCircle(x, y, 15, paint);
    for (int i = 0; i < 18; i++) {
        x2 = (float) (x + 50 * Math.sin( (float) i /18 * 2*Math.PI));
        y2 = (float) (y + 50 * Math.cos( (float) i /18 * 2*Math.PI));
        bitmapCanvas.drawLine(x, y, x2, y2, paint);
    }

    invalidate();
    return true;
}

```

Umwandeln zum Zeichenbrett

Sonst wird onDraw nicht aufgerufen



# OnTouchListener

```
package eh.beispiele.canvas;

import eh.beispiele.canvas.view.SonnenscheinDraw;

public class SonnenscheinActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(new SonnenscheinDraw(this));
    }
}
```

Viewdefinition ohne XML

# ein Touch in Google-Maps

## Optionsmenü

```

case R.id.opt_ort:
    /* neue Schicht an sich ohne weitere Funktion,
     * allerdings wird dadurch auch das TouchEvent überschrieben.
     * Damit wird die Koordinatenübernahme möglich. */
    mapOverlay = new MapOverlay();
    mMapView.getOverlays().add(mapOverlay);
    mMapView.invalidate(); // aktualisieren
    return true;

```

```

class MapOverlay extends com.google.android.maps.Overlay {

/* Überschreibt das TouchEvent der Klasse MapOverlay
 * Wenn nun an das Display berührt wird, werden die Koordinaten abgelesen,
 * in Koordinaten umgerechnet und an neupoi übergeben. */
@Override
public boolean onTouchEvent(MotionEvent event, MapView suchmapview) {
    if (event.getAction() == 1) { // Touchpad loslassen
        GeoPoint poi = suchmapview.getProjection().fromPixels((int) event.getX(), (int) event.getY());
        final Intent intent = new Intent();
        intent.putExtra(MapsPoILG, poi.getLongitudeE6());
        intent.putExtra(MapsPoIBG, poi.getLatitudeE6());
        setResult(Activity.RESULT_OK, intent);
        finish();
    }
    return false;
}
}

```

ACTION\_DOWN = 0  
ACTION\_UP = 1

was anderes interessiert nicht

# SMS versenden

I

N

Manifest:

F

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

O

Activity:

R

```
final SmsManager smsManager = SmsManager.getDefault();
```

M

```
smsManager.sendTextMessage(telstr, null, smstext, null, null);
```

A

T

I

K

# SMS empfangen

I

N

Manifest:

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

F

Activity:

O

```
private static final String SMS_EMPFANGEN_ACTION =
    "android.provider.Telephony.SMS_RECEIVED";
```

R

onCreate:

nach Intents lauschen

M

```
registerReceiver(new BroadcastReceiver() {
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(SMS_EMPFANGEN_ACTION)) {
            textSMS(context, intent);
        }
    }
}, new IntentFilter(SMS_EMPFANGEN_ACTION));
```

falls was kommt,

und wenn es eine SMS ist,

A

dann tu was damit ...

T

so wird es vereinbart

I

Die Komponente, die die SMS empfängt, sendet anschließend den Broadcast in einem Intent mit der Bezeichnung `android.provider.Telephony.SMS_RECEIVED` aus. Mit

K

```
Intent intent = new Intent(Intent.ACTION_TIME_CHANGED);
sendBroadcast(intent);
```

kann man selbst einen Broadcast an alle anderen Apps senden (in diesem Fall die Nachricht, dass die Uhrzeit geändert wurde).

# SMS empfangen

empfangener, fremder Intent wird verarbeitet

```
private void textSMS(final Context context, final Intent intent) {
    Bundle bundle = intent.getExtras();
    if (bundle != null) {
        Object[] pdus = (Object[]) bundle.get("pdus");
        SmsMessage[] msgs = null;
        String str = "";
        msgs = new SmsMessage[pdus.length];
        for (int i=0; i<msgs.length; i++){ // alle SMS
            msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
            String daten = msgs[i].getMessageBody().toString();
            String pre = daten.substring(0, 3);
        }
    }
}
```

protocol discription unit,  
Norm für den Aufbau einer SMS

alle SMS in ein array

SMS als  
array of byte

daraus die Textnachricht extrahieren

...

und weiterverarbeiten

# Anbindung von Datenbanken

I

N

F

O

R

M

A

T

I

K

- Android verfügt auch über eine eigene einfache Datenbank (SQLite)
- Android (java.sql.\*) kann auf externe DB direkt zugreifen
- Besser und sicherer ist der Zugriff über ein Webinterface (php-Scripte auf einem Webserver)
  - Keine Zugriffsdaten zur DB außerhalb des Webservers
  - Bei vielen Änderungen/Optimierungen an der DB muss keine Änderung der App erfolgen, sondern nur die php-Scripte angepasst werden.
- In der Regel sind XAMPP-Kenntnisse vorhanden, so dass auf diesem Wege die Neuerungen minimiert werden.

# Anbindung von Datenbanken

I  
N  
F  
O  
R  
M  
A  
T  
I  
K

```
1: app-alle.php
1 <?php
2 #####P
3 ##·Online-Raumbuchung·v1.0·····##P
4 ##·Januar·2012·····##P
5 ##·von·····##P
6 ##·Joachim·Ehlers·····##P
7 ##·joachim@ehlers-web.de·····##P
8 #####P
9 include·'strukdata.php';P
10 $today·=·date('Y-m-d');P
11 if·(isset($_REQUEST['pid']))·{$_pid·=·$_REQUEST['pid'];}·else·{$_pid·=·"";};P
12 P
13 ###·Datum·Daten·aus·DB·einlesenP
14 include·'dbdata.php';P
15 $link·=·mysql_connect($_server,$_benutzer,$_pw);P
16 mysql_select_db($_db,$link);P
17 P
18 ###·Daten·aus·DB·einlesen·(für·den·gewählten·Tag)P
19 $ergebnis·=·mysql_query("SELECT·*·FROM·belegung·WHERE·(datum·>=·'$_today')·and·(pers·=·'$_pid')·order·by·datum,·std");P
20 $anz·=·mysql_num_rows($ergebnis);P
21 while·(($datensatz·=·mysql_fetch_row($ergebnis))P
22 ··$buch[]·=·$datensatz;P
23 mysql_close($link);P
24 P
25 if·($anz·>·0)·{P
26 ··foreach·($buch·as·$zeile)·{P
27 ···print·$zeile[1]."\n";P
28 ···print·$zeile[3]."\n";P
29 ···print·$stdbez[$zeile[2]-1]."\n";P
30 ··}P
31 P
32 else·{P
33 ··print·"nix\n";P
34 P
35 >?>
```

```
$_pid·=·$_REQUEST['pid'];
```

```
{$_pid·=·$_REQUEST['pid'];}·else·{$_pid·=·"";};
```

```
print·$zeile[1]."\n";
```

# Anbindung von Datenbanken

```

ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
try{
    nameValuePairs.add(new BasicNameValuePair("pid",pid));

    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost("http://www.beispiel.webserver.de/demo/app.php");
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    HttpResponse response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();
    InputStream ins = entity.getContent();
    BufferedReader reader = new BufferedReader(new InputStreamReader(ins, "iso-8859-1"), 8);
    String line = null;
    line = reader.readLine(); // nix oder Tripel

    if (line.compareTo("nix") == 0) {
        out = "keine Räume belegt";
    }
    else {
        out = "Datum      Raum      Std.\n";
        out += getdatum(line) + ": ";
        line = reader.readLine();
        // ... usw. ...
    }

    ins.close();
}
catch(Exception e){
    Log.e("log_tag", "Fehler: "+e.toString());
    out = e.toString();
}

```

**Anfrage** (red arrow pointing to the `HttpPost` object creation)

**Antwort** (green arrow pointing to the `HttpResponse` object)

Adresse des php-Scriptes

Übergabeparameter

Standard für die Zeichenkodierung



# Der Beschleunigungssensor

Activity:

```
public class shake extends Activity implements SensorEventListener {  
    private SensorManager mSensorManager;  
    ...  
}
```

alle Methoden überschreiben!

onCreate:

```
mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);  
mSensorManager.registerListener(this,  
    mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),  
    SensorManager.SENSOR_DELAY_NORMAL);
```

Beschleunigungssensor

# Der Beschleunigungssensor

alle Methoden überschreiben!

```
@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        datenzeigen(event);
    }
}
```

auch wenn nicht relevant sind ...

```
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
```

```
float[] values = event.values;
float x = values[0];
float y = values[1];
float z = values[2];
```

# Der Beschleunigungssensor

I

N

F

O

R

M

A

T

I

K

Was nicht benötigt wird, wird ausgeschaltet.  
Neuregistrierung nicht vergessen!

```
@Override
protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
        SensorManager.SENSOR_DELAY_NORMAL);
}

@Override
protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
}
```

# Übung

I

N

Vorschläge:

F

- singuläre Positionsüberwachung per SMS

O

- Tracking mit Hilfe einer Online-Datenbank

R

- Erweiterung der Maps-App um OnTouch Ortswahl

M

- Schiebepuzzle

M

- Wasserwaage

A

- Balance-Spiel

T

- ...

I

K