

# Senzori

## Mobilne aplikacije

# Fizičke veličine

- Fizičke veličine opisuju svojstva materije i fizičkih pojava
- One mogu biti skalarne (temperatura vazduha, vlažnost vazduha, vazdušni pritisak), vektorske (pozicija, brzina, ubrzanje), itd.

# Merenje



Figure 1: Merenje.

- Merenje je proces upoređivanja nepoznate fizičke veličine sa poznatom fizičkom veličinom
- Postoji standardna merna jedinica za svaku fizičku veličinu
- Postoje osnovne merne jedinice (dužina, masa, vreme, električna struja, temperatura, količina supstance i jačina svetlosti) i izvedene merne jedinice

# Senzori

- Senzor je uređaj koji pretvara jednu fizičku veličinu u drugu fizičku veličinu koju čovek može neposredno da opazi (ili koju računar može da očita)

# Digitalizacija

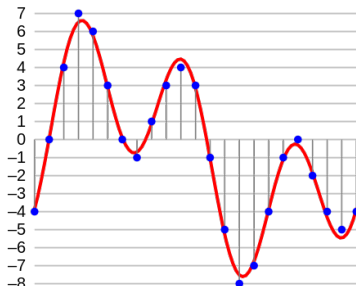
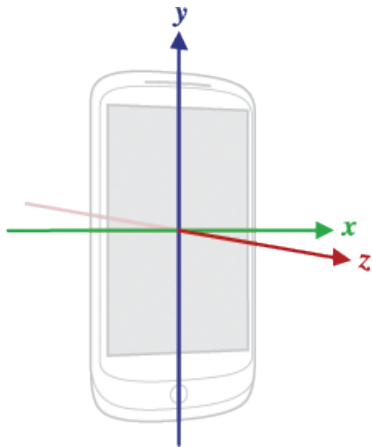


Figure 2: Digitalizacija.

- uzorkovanje (očitavanje vrednosti analognog signala (obično sa konstantnom frekvencijom))
- kvantizacija (aproksimacija očitane vrednosti sa vrednostima iz konačnog skupa)

# Senzorski koordinatni sistem



- x osa (horizontalna, od levo prema desno)
- y osa (vertikalna, od dole prema gore)
- z osa (od uređaja)

Figure 3: Senzorski koordinatni sistem.

# Senzori

Tip	Opis
ACCELEROMETER	Meri ubrzanje uređaja (sa g)
AMBIENT_TEMPERATURE	Meri temperaturu vazduha
GRAVITY	Meri g
GYROSCOPE	Meri ugaonu brzinu uređaja
LIGHT	Meri jačinu svetlosti
LINEAR_ACCELERATION	Meri ubrzanje uređaja (bez Zemljinog g)
MAGNETIC_FIELD	Meri jačinu magnetnog polja
PRESSURE	Meri vazdušni pritisak
PROXIMITY	Meri udaljenost objekta od ekrana
RELATIVE_HUMIDITY	Meri relativnu vlažnost vazduha
ROTATION_VECTOR	Meri orijantaciju uređaja

Table 1: Tipovi senzora.

# Senzori

Klasa/Interfejs	Opis
SensorManager	Omogućava pristup senzorima
Sensor	Sadrži informacije o svojstvima određenog senzora
SensorEvent	Događaj koji sadrži informacije o određenom merenju
SensorEventListener	Sadrži obrađivače SensorEvent događaja

Table 2: Sensors API.



# Senzori

- ➊ Zatražiti prava pristupa (statički ili dinamički)
- ➋ Odrediti koji senzori su dostupni na uređaju
- ➌ Odrediti mogućnosti dostupnih senzora
- ➍ Napisati obrađivače događaja koji reaguju na promenu fizičke veličine ili tačnosti merenja
- ➎ Registrovati i odregistrovati obrađivače događaja

# ExampleActivity.java

```
1 public class ExampleActivity extends Activity, implements SensorEventListener {
2
3     private SensorManager sensorManager;
4
5     public onCreate() {
6         // ...
7         sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
8
9         // Lists all sensors on a device
10        List<Sensor> sensors = sensorManager.getSensorList(Sensor.TYPE_ALL);
11
12        // Lists all sensors of a given type
13        List<Sensor> sensors = sensorManager.getSensorList(Sensor.TYPE_MAGNETIC_FIELD);
14
15        // Determines whether a specific type of sensor exists
16        if (sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null) {
17            // Success! There's a magnetometer.
18        } else {
19            // Failure! No magnetometer.
20        }
21    }
22    // ...
23 }
24
```

# Senzori

Metoda	Opis
<code>float getMaximumRange()</code>	maksimalan raspon izmerenih vrednosti
<code>int getMinDelay()</code>	minimalan period između dva merenja
<code>float getResolution()</code>	rezolucija senzora
<code>float getPower()</code>	potrošnja

Table 3: Metode klase Sensor.

# Senzori

Metoda	Opis
<code>String getName()</code>	ime senzora
<code>int getType()</code>	generički tip senzora
<code>String getVendor()</code>	proizvođač senzora
<code>int getVersion()</code>	verzija senzora

Table 4: Metode klase Sensor.

# ExampleActivity.java

```
1 public class ExampleActivity extends Activity, implements SensorEventListener {  
2  
3     private final SensorManager sensorManager;  
4     private Sensor sensor;  
5  
6     // ...  
7  
8     public void onAccuracyChanged(Sensor sensor, int accuracy) {  
9         // Called when the accuracy of a sensor has changed.  
10    }  
11  
12    public void onSensorChanged(SensorEvent event) {  
13        // Called when sensor values have changed.  
14    }  
15 }  
16
```

# Senzori

Metoda	Opis
<code>onSensorChanged(SensorEvent event)</code>	Obađuje promenu fizičke veličine
<code>onAccuracyChanged(Sensor sensor, int accuracy)</code>	Obraduje promenu tačnosti merenja

Table 5: Metode interfejsa `SensorEventListener`.

# Senzori

Atribut	Opis
<code>float[] values</code>	izmerena vrednost (skalar ili vektor)
<code>long timestamp</code>	vreme merenja [ns]
<code>int accuracy</code>	tačnost merenja
<code>Sensor sensor</code>	korišćen senzor

Table 6: Atributi klase SensorEvent.

# ExampleActivity.java

```
public class ExampleActivity extends Activity , implements
    SensorEventListener {

2
    private final SensorManager sensorManager;
4    private Sensor sensor;

6    // ...

8    protected void onResume() {
        super.onResume();
10    sensorManager.registerListener(
        this , sensor , SensorManager.SENSOR_DELAY_NORMAL);
12    }

14    protected void onPause() {
        super.onPause();
16    sensorManager.unregisterListener(this);
    }

18    // ...
20 }
```



# Senzori

Parametar	Opis
<code>listener</code>	obrađivač događaja
<code>sensor</code>	senzor
<code>samplingPeriodUs</code>	period uzorkovanja

Table 7: Parametri metode `registerListener`.

# Dobra praksa

- Koristiti Google Play filtere za izbor uređaja sa odgovarajućim tipovima senzora ili detektovati senzore u toku izvršavanja aplikacije i po potrebi o(ne)mogućiti određene funkcije
- Odregistrovati obrađivač događaja kada senzor više nije potreban (štedi bateriju)