



# Mobile Technologies & Smart Devices

2020~2021

**overtref jezelf**



# Huiswerk

- ▶ Wat is een Exception?

# Try en catch

- ▶ Soms kunnen er op runtime dingen mis gaan
- ▶ Er wordt dan een exception gegenereerd die ervoor zorgt dat de uitvoering van het programma gestopt wordt
- ▶ Als we bepaalde code “wantrouwen” dan kunnen we deze binnen een try-catch blok zetten.  
Als de code binnen try lukt wordt catch overgeslagen, als try faalt dan wordt de code binnen catch uitgevoerd, met als parameter de exception

```
try
{
    // deel van de code waar iets mis kan gaan
    c = a / b;
}
catch (Exception e)
{
    System.out.println("Code faalt: " + e.Message);
}
```

# Try en catch

# Try en catch (vervolg)

- ▶ Exception is de basis class waar alle andere exceptions vanaf stammen
- ▶ Als je al weet wat er mis kan gaan kan je in je catch dus een specifieker exception opvangen
- ▶ Met meerdere catch' en kan je verschillende fouten, verschillend afhandelen

```
try
{
    // open bestand
    // probeer xml in te lezen en vertaal naar json
    // schrijf bestand als json weg
}
catch (OpenFileException e)
{
    // laat weten dat bestand niet geopend kan worden
}
catch (IncorrectXMLException e)
{
    // laat weten dat formaat niet klopt
}
catch (WriteFileException e)
{
    // laat weten dat het niet opgeslagen kan worden
}
```

# Eigen Exceptions

- ▶ Je kunt ook zelf exceptions genereren

```
if (n == 0)
{
    throw new NeverUseZeroException();
}
return (m / n);
```

- Als er tijdens runtime iets mis kan gaan waar geen voorgedefinieerde exception voor bestaat, dan kan je je eigen exception maken door de class Exception uit te breiden

# Internet

- ▶ Toestemming
- ▶ Nooit op de UI thread

# Internet

# Internet

- ▶ Permissie: INTERNET en ACCESS\_NETWORK\_STATE
- ▶ HttpURLConnection (Java, niet Android specifiek)
- ▶ Verboden op main -> onderbrengen in AsyncTask die op asynchroon/op de achtergrond draait (doInBackground)
- ▶ Deze start je door een nieuwe instantie van de taak aan te maken, en daarvan de execute() aan te roepen
- ▶ Callback (onPostExecute) geeft de info terug vanuit de AsyncTask naar de main Activity (aanpassen UI is verboden vanuit achtergrondtask)

<https://developer.android.com/training/basics/network-ops/connecting.html>

# Internet

- ▶ Volley library
- ▶ Handelt request voor je af
- ▶ Heeft eigen mechanisme om op de achtergrond te draaien

<https://developer.android.com/training/volley>

# Collections

- ▶ OO Gegevensverzamelingen
- ▶ Grootte is flexibel => wordt dynamisch voor je aangepast

# Collections: typen

- ▶ Er zijn erg veel verschillende Collections
- ▶ Bijv: ArrayList, HashMap, Set, Queue, Stack

Collection  
Iterable

# Collections: typed

- ▶ Bij sommige collections moet je van tevoren aangeven welk type data erin gaat, bij andere hoeft dit niet
- ▶ ArrayList kan zowel typed als untyped gebruikt worden

```
ArrayList<String> strings = new ArrayList<>();
strings.add("test");
strings.add(1); // ERROR!
```

```
ArrayList all = new ArrayList();
all.add("test");
all.add(1);
```

# Collections: Lists

- ▶ Volgorde is belangrijk
- ▶ ArrayList
- ▶ Snel opzoeken op nummer (index)

Sort, Reverse

# Collections: NameValueCollections

- ▶ Index based accessing (Dictionaries)
- ▶ HashMap<Tkey,Tvalue>
  
- ▶ Snel opzoeken op key
- ▶ Snel opzoeken of key bestaat

# Collections: Sets

- ▶ Geen dubbele waarden
- ▶ Set<T>
- ▶ Snel checken of iets erin zit

# Collections: Queue en Stack

- ▶ LIFO en FIFO
- ▶ Add en remove / push en pop
- ▶ Peek
- ▶ Queue<T>, Stack<T>

Genereren een exception als er niks meer “te halen” valt

# Lijstgegevens

# Lijstgegevens

- ▶ Gegevens koppelen aan Adapter
- ▶ Adapter koppelen aan view (listView, scrollView)
- ▶ Met onItemClickListener aanklikbaar te maken
- ▶ Update via adapter

```
// Lijstgegevens
names = new ArrayList();
Names.add("Naam 1");

// Aan adapter koppelen (default views gebruikt, maar eigen kunnen ook)
myAdapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1,
    android.R.id.text1, names);

// Koppel aan lijst
myListView.setAdapter(myAdapter);

// aanpassing in Array? Dan update lijst via adapter
myAdapter.notifyDataSetChanged()
```

<http://developer.android.com/guide/topics/ui/layout/listview.html>

# Workshop

- ▶ Huiswerk soundboard
- ▶ Oefenen met internet en lijsten
  - ▶ Volley
- ▶ Vergeet niet te werken aan je praktijkopdracht

# Multimedia: Audio

- ▶ RAW audio-bestand in resource (/res/raw)
- ▶ SetVolumeControlStream, Volume knop aanpassen
- ▶ Medioplayer doet het werk
- ▶ Voorbeeld soundboard

```
// Constructor Activity  
setVolumeControlStream(AudioManager.STREAM_MUSIC);  
  
// in Listener, ofzo (mp is instantie var)  
if (mp!=null) { mp.release(); }  
mp = MediaPlayer.create(this, R.raw.hastalavista);  
mp.start();
```

# Multimedia: Video

- ▶ Video komt van geheugen (bij voorkeur sd)
- ▶ Emulator: Sdcard aanmaken / AVD manager
- ▶ Video heeft videoview nodig

```
<VideoView  
    android:id="@+id/video"  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:layout_gravity="center" />
```

```
VideoView video = (VideoView) findViewById(R.id.video);  
video.setVideoPath("/sdcard/samplevideo.3gp");  
video.start();
```