

Week 5 - oefening 1: Bitcoin

Doelstelling

In deze opgave bekijken hoe één fragment met twee fragments onderling met elkaar communiceren, alsook het gebruik van de backstack, het gebruik van het Bundle-object, SharedPreferences, enz.

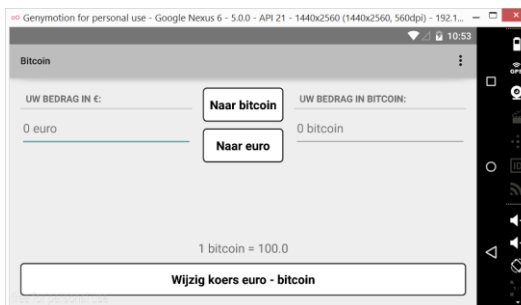
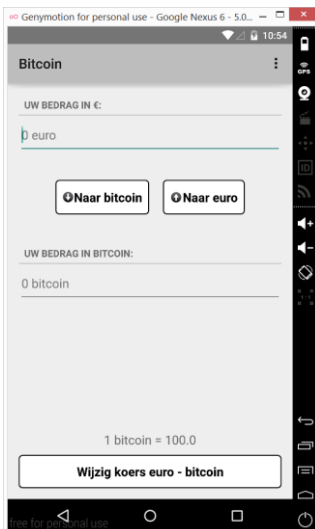
Voorbereiding

Maak een nieuw Android Application Project aan met volgende instellingen.

Application Name:	Bitcoin
Package Name:	be.howest.nmct
Target SDK:	21
Activity:	MainActivity – <i>layout: activity_main.xml</i>
Fragments:	ChangeFragment - <i>layout: fragment_change.xml, layout-land\ fragment_change.xml</i> BitcoinRateFragment – <i>layout: fragment_bitcoinrate</i>
Launch icon:	Zie bijlage

Design

Ontwikkel volgend design: dit bestaat uit 2 fragments: ChangeFragment en BitcoinRateFragment. Voor het eerste fragment voorzien we ook een landscape-versie.





Opmerkingen:

Bovenstaand design werd via een RelativeLayout gerealiseerd. De twee centrale buttons staan in een LinearLayout.

Een afbeelding samen met tekst in een button plaatsen kan via een gewone button waarbij het xml-attribuut `drawableLeft` de afbeelding links van de tekst plaatst.

`android:drawableLeft="@drawable/arrow_down"`

De style van de button werd bekomen door:

1. mijn_button.xml in de map drawable:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#FFFFFF" />
    <corners android:radius="5dp" />
    <stroke android:width="5px" android:color="#000000" />
</shape>
```

2. In het bestand Styles aanvullen:

```
<style name="CustomStyleButton" parent="@android:style/Widget.Button">
    <item name="android:textSize">18sp</item>
    <item name="android:textStyle">bold</item>
    <item name="android:gravity">center</item>
    <item name="android:background">@drawable/mijn_button</item>
</style>
```

3. In de layoutfile van de activity:

```
<Button
    android:id="@+id/btnWijzigen"
    style="@style/CustomStyleButton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editTextKoersEuro"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:onClick="changeRate"
    android:text="@string/save" />
```

Code - Voorbereiding

Maak voor elk fragment een bijhorende Fragment-klasse: ChangeFragment, BitcoinRateFragment.

ChangeFragment:

1. Implementeer de onCreateView-methode en haal alle relevante control-views op
2. Maak op moduleniveau een float-attribuut 'currentRateBitcoinInEuro' aan dat de wisselkoers van 1 bitcoin in euro's voorstelt.
3. Zorg dat onderaan steeds de wisselkoers weergegeven wordt. Voorzie hiervoor een afzonderlijke methode.
4. Maak de private methodes 'changeToEuro' en 'changeToBitcoin' aan dat resp. de juiste waarde ophaalt, de berekening met de wisselkoers uitvoert en het resultaat op de juiste plaats afprint. Zorg dat deze twee methodes via de juiste button geactiveerd worden.

BitcoinRateFragment

1. Implementeer de onCreateView-methode en haal alle relevante control-views op
2. Maak op moduleniveau een float-attribuut 'rate1BitcoinInEuros' aan dat de wisselkoers van 1 bitcoin in euro's voorstelt.
3. Maak een methode dat de nieuwe wisselkoers ophaalt én voorlopig in de modulevariabele bewaard.

Code – Wisselen van fragment in Activity

Bestudeer de uitleg op onderstaande link.

<http://developer.android.com/guide/components/fragments.html#Transactions>

Voorzie in de **MainActivity** 2 methodes die elk instaan voor het wisselen van getoonde fragment naar hun bijhorende fragment:

- showFragmentBitcoinRate
- showFragmentChange

Baseer je op het voorbeeld op bovenstaande link.

Opmerking: in het voorbeeld werkt men met de default-constructor. De default-constructor is essentieel en moet steeds voor handen zijn (waarom?). Wanneer je echter data vanuit de activity wenst door te sturen naar een Fragment, zullen we op een andere manier moeten werken, gebruik makend van deze constructor:

- Voorzie een static-methode 'newInstance' dat een object van de Fragment aflevert. Voorzie de gewenste parameters
- Maak gebruik van de Bundle-klasse om de parameters bij te houden. Een bundle werkt met key-value-paren. Gebruik voor de keys steeds String-constanten.
- Stel de argumenten van de Fragment in.

```
static final String BITCOIN_RATE = "be.howest.nmct.NEW_BITCOIN_RATE";

public static BitcoinRateFragment newInstance(float bitcoinrate) {
    BitcoinRateFragment fragment = new BitcoinRateFragment();
    Bundle args = new Bundle();
    args.putFloat(BITCOIN_RATE, bitcoinrate);
    fragment.setArguments(args);
    return fragment;
}

//noodzakelijk
public BitcoinRateFragment() {
}
```

In de onCreate-methode kunnen de parameters opgevraagd worden.

```
//data binnengekregen vanuit activity
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        rateBitcoinInEuros = getArguments().getFloat(BITCOIN_RATE);
    }
}
```

Pas dit nu toe:

We wensen vanuit de activity de huidigeWisselkoers naar beide Fragments doorte geven. Maak eerst in de activity een modulevariabele (float) voor de recente wisselkoers bij. Geef deze voorlopig een default waarde.

Voeg in beide Fragment-klassen de static methode 'newInstance' met float-parameter toe. Voeg eveneens de onCreate-methode toe waarin de parameters opgevraagd worden. Met de doorgegeven waarde wordt in beide klassen de modulevariabele ingesteld.

Pas nu de methodes showFragmentBitcoinRate en showFragmentChange aan zodat de recente wisselkoers via de methode newInstance vanuit de Activity kan doorgegeven worden

Code – communicatie Fragment > Activity

De communicatie vanuit Fragment naar Activity wordt geïllustreerd op onderstaande link:

<http://developer.android.com/training/basics/fragments/communicating.html>

Pas deze manier toe om voor volgende zaken

- aanvraag om fragment BitcoinRateFragment te tonen.
In ChangeFragment kan de gebruiker op de button "Wijzig koers" klikken. De activity moet daardoor getoonde changefragment wisselen met BitcoinRateFragment. Roep eerder geschreven methode 'showFragmentBitcoinRate' aan.
- aanvraag om fragment changeFragment te tonen.
In BitcoinRateFragment kan de gebruiker een nieuwe koers opgeven en klikken op de button. De activity moet het getoonde BitcoinRateFragment wisselen met ChangeFragment. Stuur via de methode ook de nieuwe koers mee. Roep eerder geschreven methode 'showFragmentChange' aan.

Stappenplan:

1. voorzien de Fragment van een interface met methode(s)
2. maak in uw fragment op moduleniveau een variabele van deze interface aan
3. in de attach-methode van de Fragment-klasse koppel je de activity-parameter aan deze variabele
4. Neem de activty en laat deze de interface uit de fragment-klasse implementeren
5. Werk de nieuwe methode in de activity verder uit.
6. Gebruik deze methode vanuit Fragment-klasse op.

Code – Activity : SharedPreferences

Gebruik opnieuw SharedPreferences-klasse om de wisselkoers te bewaren wanneer de app beëindigd wordt.

<http://developer.android.com/guide/topics/data/data-storage.html>