

# RxJava

Jan Staněk

ANDROID DEVELOPER @ ACKEE

#CODECAMP



# Obsah

- Imperativní vs. funkcionální vs. reaktivní programování
- RxJava + Observable pattern
- Lambda výrazy + reference na metody
- Operatory
  - Create, From, Just, Defer, Range
  - Filter, Map, Take/Last/First, Distinct
  - MathObservable, Scan
  - Merge
  - Zip
  - CombineLatest
- Schedulers
- Retrofit, RxBinding, SqlBrite

# Imperativní vs. funkcionální vs. reaktivní programování

```
// Imperativní přístup
```

```
String[] strings = new String[]{  
    "první", "druhy", "třetí", "čtvrtý", "pátý"  
};
```

```
int totalLength = 0;  
for (int i = 0; i < strings.length; ++i) {  
    totalLength += strings[i].length();  
}
```

```
//-----
```

```
// Funkcionální přístup
```

```
Observable.from(strings)  
    .map(String::length)  
    .reduce((integer, integer2) -> integer + integer2)  
    .subscribe(sum -> {  
        // ....  
    });
```

# Imperativní vs. funkcionální vs. reaktivní programování

## Reaktivní programování

- Funkcionální přístup
- Asynchronní
- Observable pattern
- Iterable pattern + push místo pull
- Množství operátorů

# RxJava + Observable + Observer

- Observer
  - onNext
  - onComplete
  - onError

Simulace observable (výrobní linky), který nám bude:

- emitovat papíry
- papíry budou dostatečně dlouhé
- papíry se obarví červeně
- červené papíry se sesbírají do krabice
- výsledek se předa observerovi

# Lambda výrazy + reference na metodu

- Java 8
- Retrolambda
- Jack kompilátor, který přichází s Androidem N
  - Podporuje Lambda výrazy
  - Reference na metodu (zatím?) nejsou Jackem backportovány

```
dependencies {  
    classpath 'me.tatarka:gradle-retrolambda:3.2.4'  
}
```

# Lambda výrazy + reference na metodu

```
Button superTlacitko = (Button) view.findViewById(R.id.btn_super);
superTlacitko.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Log.i(TAG, "FUJ!");
    }
});
```

```
// Lambda
superTlacitko.setOnClickListener(v -> Log.i(TAG, "uuu nadhera"));
```

```
// Method reference
superTlacitko.setOnClickListener(this::onSuperTlacitkoClicked);
```

```
// Method reference
superTlacitko.setOnClickListener(Logger::logClick);
```

```
}

public static class Logger {
    public static void logClick(View v) {
        Log.d("Logger", "Clicked whohooo");
    }
}
```

```
public void onSuperTlacitkoClicked(View v) {
    Log.i(TAG, "Mega super!");
}
```

# Vytváření observable

- Just
- From
- Create
  
- Defer
- Range

Ukázka



# Základní operátory

- Filter
- Map
- Take/Last/First
- Distinct
- Max
- Min

Ukázka

<https://github.com/ReactiveX/RxJavaMath>

# Operátory na práci s vlákny aneb smrt asyncTaskům

- defaultně Observable s řetězem operátorů vykonají práci na threadu, na kterém se zavola subscribe.
- operátory na práci s vlákny
  - SubscribeOn
    - Nastaví, na kterém vlákně se má začít pracovat.
  - ObserveOn
    - Nastaví, na kterém vlákně se má pokračovat dále v chainu. Může se použít opakovaně.
    - RxAndroid přichází s `AndroidSchedulers.mainThread()`
- Vlákna k dispozici
  - `Schedulers.io()`
  - `Schedulers.computation()`
  - `Schedulers.newThread()`
  - ...

# Kombinující operátory

- Merge
- Zip
- CombineLatest

# Kombinující operátory : Merge

## Merge

combine multiple Observables into one by merging their emissions



merge



# Kombinující operátory : Zip

## Zip

combine the emissions of multiple Observables together via a specified function and emit single items for each combination based on the results of this function

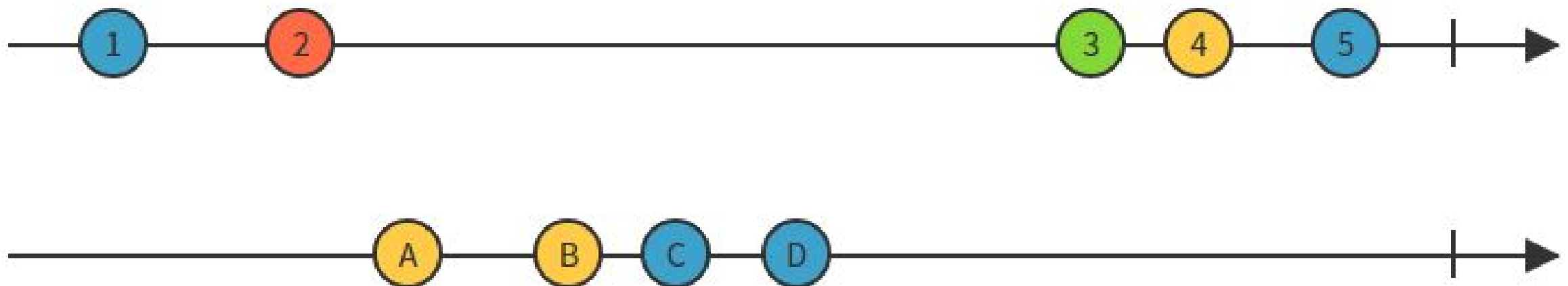


zip



## CombineLatest

when an item is emitted by either of two Observables, combine the latest item emitted by each Observable via a specified function and emit items based on the results of this function



```
combineLatest((x, y) => "" + x + y)
```



# RxJava + Retrofit

Slast!

Ukázka

compile "com.squareup.retrofit2:adapter-rxjava2.0.2"

# Odkazy na další počtění a více informací

<https://github.com/ReactiveX/RxJava/wiki>

<http://reactivex.io/tutorials.html>

<http://rxmarbles.com/>

<https://github.com/square/sqlbrite/>

<https://github.com/JakeWharton/RxBinding>

Poslech

<http://fragmentedpodcast.com/episodes/3/>

<http://fragmentedpodcast.com/episodes/4/>





JAN STANĚK

jan.stanek@ackee.cz