

# Übung

## Lambdas und Functional Interfaces

### 1 Die Aufgabe

Programmiere ein funktionales Interface mit dem Namen `NumberSelector`, dessen Methode *check* einen `int`-Wert als Parameter hat und `boolean` returnt.

Implementiere in `NumberSelector` eine statische Methode *findFirst*, die als Parameter einen `NumberSelector` und eine `int`-Zahl hat.

Die Methode soll die ersten  $n$  `int`-Werte  $n > 0$  und  $n < 1.000.000$  finden, für die der Selector `true` zurückgibt und diese als `int[]`-Array zurückgeben.

**Folgendes soll gefunden werden:**

- die ersten 10 Zahlen, die durch 7 teilbar sind
- die ersten 10 Primzahlen  
 $(p_n)_{n \in \mathbb{N}} = (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, \dots)$
- die Zahlen von 0 bis 20

Ein functional Interface gibt es in Java seit der Version 8 (seit den Lambdas). Dieses Interface besitzt nur **eine** abstrakte-Methode **nicht** implementierte Methode. Dieses Interface kann noch zusätzliche statische implementierte und default-Methoden besitzen. Sie besitzen außerdem die Annotation `@FunctionalInterface`. Solche Schnittstellen werden von üblicherweise von Lambda-Expressions implementiert.

## Ein Beispiel

```
@FunctionalInterface
public interface MyFunctionalInterface {
    void execute();
}
```

## Merkhilfe für die Lambda-Syntax

$$(\text{LambdaParameter}) \rightarrow \{\text{Anweisungen}\}$$