ToiletPaper #135

Enum Reverse Lookup in Kotlin

Autor: Andreas Hofmann / Software Engineer / Standort Stuttgart

✗ Problem

Kotlin bietet die Möglichkeit, einem Aufzählungstypen Aufzählungswerte zuzuweisen:

```
enum class Currency(val symbol: String) {
    DOLLAR("$"),
    EURO("€"),
    PFUND("£");

    override fun toString(): String {return this.symbol}
}
```

So kann durch Auswahl des Aufzählungstyps dessen Wert zurückgegeben werden:

```
Currency.valueOf("DOLLAR") //  liefert den Aufzählungswert "$" zurück
```

Möchte man aber ausgehend von einem Wert auf den entsprechenden Eintrag schließen, bietet Kotlin dafür keine native Funktion an. Man muss also über alle Aufzählungstypen iterieren und jeweils den Wert mit dem gesuchten Wert vergleichen:

```
Currency.values().firstOrNull {currency -> currency.symbol == "$"} //  liefert den Aufzählungstyp
                                                                       DOLLAR zurück
```

Diese Art von Lösung findet sich im Netz in der einen oder anderen Variante und sie tut, was sie soll. Aber möchte man jedes Mal über alle Aufzählungen iterieren und Werte vergleichen? Gibt es dafür keine elegantere Lösung?

✓ Lösung

Wir erweitern unsere Aufzählungsklasse mit einem Companion Object:

```
companion object {
    private val mapping = values().associateBy(Currency::symbol)
    fun fromSymbol(symbol: String) = mapping[symbol]
}
```

Die Funktion associateBy() liefert eine Map<K, T> zurück. Genauer eine LinkedHashMap bei der K der Aufzählungswert und T unser Aufzählungstyp ist. Die Funktion "fromSymbol()" nimmt den Aufzählungswert als Argument entgegen und liefert uns den passenden Aufzählungstyp. Der Zugriff auf den Aufzählungstyp über den Aufzählungswert erfolgt jetzt nicht mehr durch iterieren in O(n), sondern über einen Hash Lookup in der statischen Map in O(1).
Mit dieser Erweiterung kann der Reverse Lookup folgendermaßen durchgeführt werden:

```
Currency.fromSymbol("$") //  liefert den Aufzählungstyp DOLLAR zurück
```

Damit haben wir eine Lösung mit leichter lesbarem Code, die u. U. sogar performanter bez. der Ausführungszeit sein kann. Vorsicht: Wenn es keine 1 zu 1 Beziehung zwischen Aufzählungstypen und Aufzählungswerten gibt, liefert diese Lösung nicht den ersten Treffer, sondern den letzten Treffer.

+ Weiterführende Aspekte

- https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/enum-value-of.html
- https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.sequences/associate-by.html