4.6 Übungen zum Date and Time API

Lernziel: Die Verarbeitung von Datumswerten wurde mit Java 8 überarbeitet und deutlich vereinfacht. Die folgenden Übungen liefern einen Einstieg in die neuen Klassen, in Berechnungen und weitere Hilfsfunktionalitäten.



Repräsentieren Sie Ihren Geburtstag als LocalDate, den Feierabend 17:30 Uhr als LocalTime und den jetzigen Moment als LocalDateTime. Starten Sie z. B. wie folgt:

```
final LocalDate myBirthday = LocalDate.of(1971, 2, 7);
```

Tipp: Experimentieren Sie mit den Methoden now(), of() und parse(String).

Berechnen Sie die Zeitdauer zwischen heute und Ihrem Geburtstag bzw. andersherum.

```
final LocalDate now = LocalDate.now();
final LocalDate birthday = LocalDate.of(1971, 2, 7);
```

Tipp: Verwenden Sie die Klasse LocalDate und die Methode until (). Probieren Sie als Variante auch den Aufruf von Period.between ().

Aufgabe 3a: Erzeugen Sie eine Zeitdauer von 1 Jahr und 2 Monaten und 14 Tagen. Schauen Sie sich die Konsolenausgaben an und variieren Sie die Werte:

```
final Period period = Period.of(1, 2, 14);
```

Aufgabe 3b: Wie lautet die Ausgabe für eine Hintereinanderschaltung von Aufrufen der Methoden ofYears (1), ofMonths (2) und ofDays (14)?

Aufgabe 3c: Kombinieren Sie die Methode ofYears() und withXYZ()-Methoden zum Erzeugen einer Period. Welche Ausgabe erhält man dann? Und warum?

Aufgabe 3d: Addieren Sie die Zeitdauer von 1 Jahr und 2 Monaten und 14 Tagen auf den 10.10.2013. Wo landet man?

🔎 Aufgabe 4 🛍__

Aufgabe 4a: Stellen Sie Berechnungen mit JDK 8 an: Springen Sie ausgehend vom 2.2.2012 und vom 2.2.2014 sowie vom 4.4.2014 und vom 5.5.2014 mit der Methode plusMonths() der Klasse LocalDate einen Monat in die Zukunft und geben Sie dieses Datum aus.

Aufgabe 4b: Was passiert, wenn man für die Berechnungen statt plusMonths () die Methode plusDays () nutzt: Was entspricht einem Monat? 28, 29, 30 oder 31 Tage? Wie ermittelt man die korrekte Länge eines Monats?

Tipp: Die Klasse Month besitzt die Methode length (boolean). Verwenden Sie alternativ auch LocalDate.lengthOfMonth(). Worin besteht der Unterschied?



Ermitteln Sie alle Zeitzonen, die mit "America/L" oder "Europe/S" starten, und geben Sie diese sortiert auf der Konsole aus oder befüllen Sie damit eine Liste.

Tipp: Nutzen Sie die Klasse ZoneId und deren Methode getAvailableZoneIds (). Setzen Sie Streams und das Filter-Map-Reduce-Framework ein, um die zu den zuvor angegebenen Präfixen passenden Zeitzonen-IDs zu ermitteln.

______ 🕰 Aufgabe 6 🔎___

Wenn ein Flug von Zürich nach San Francisco 11,5 Stunden dauert und am 7.7.2014 um 14:30 Uhr startet, zu welcher Ortszeit landet man dann in San Francisco? Welcher Zeit entspräche das am Abflugsort? Beginnen Sie die Berechnungen wie folgt:

Tipp: Nutzen Sie die Klasse Duration und deren Methoden ofHours() sowie plusMinutes(), um die Flugzeit zu modellieren. Die Zeitzone in San Francisco ist "America/Los_Angeles". Verwenden Sie die Methoden plus() sowie withZone-SameInstant() aus der Klasse ZonedDateTime.



Aufgabe 7a: Welcher Wochentag war der Heiligabend 2013 (24.12.2013)? Was für ein Wochentag waren der erste und der letzte Tag im Dezember? Starten Sie mit folgenden Programmzeilen:

```
final LocalDate christmasEve = LocalDate.of(2013, 12, 24);
System.out.println(DayOfWeek.from(christmasEve));
```

Aufgabe 7b: Berechnen Sie das jeweilige Datum des ersten und letzten Freitags und Sonntags im März 2014. Starten Sie mit folgendem Sourcecode-Schnipsel:

Tipp: Nutzen Sie dabei die Klasse TemporalAdjusters und ihre Hilfsmethoden und die Methode with () aus der Klasse LocalDate.

Aufgabe 7c: In Aufgabe 7b haben wir vier Datumswerte im März 2014 ermittelt. Der wievielte Tag im März war das jeweils? Und der wievielte im Jahr?

Tipp: In der Klasse LocalDate findet man die Methoden getDayOfMonth() sowie getDayOfYear().