

## AUFGABE: SWIFTNESS

**DIE FOLGENDEN AUFGABEN LASSEN SICH AM EFFIZIENTESTEN IM PLAYGROUND LÖSEN.  
FÜR ALLE AUFGABEN SOLLEN EIGENE TEST-DATEN UND –AUFRUFE GENERIERT WERDEN, UM DIE  
LAUFFÄHIGKEIT UND KORREKTHEIT DER APPLIKATIONEN ZU DEMONSTRIEREN.**

### Teilaufgabe 1

Entwickeln Sie eine Swift-Funktion, die die Plausibilitätsprüfung von angenommenen Neufahrzeugen aus der Produktion realisiert. Die Fahrzeuge können entweder LKWs oder PKWs sein. Zu den PKWs gehört eine 12-Stellige Seriennummer bestehend aus Buchstaben und Zahlen. Zu den LKWs gehören zusätzlich noch die Achszahl und das zugelassene Gesamtgewicht.

Folgende Plausibilitäten sollen durch die Plausibilitätsfunktion geprüft werden:

- Die Seriennummer der PKWs darf nicht die Buchstabenfolge „WBABA“ beinhalten.
- Bei den LKWs darf die Last der einzelnen Achsen nicht 2 T überschreiten.

Die LKWS und PKWs sollen als enumeration-Typ realisiert werden.

### Teilaufgabe 2

Entwickeln Sie eine Klasse „Mitarbeiter“. Die Klasse hat die (Objekt-)Attribute: Stundenzahl, Name, Vorname, Mitarbeiternummer und Anstellungsdatum. Sie soll zusätzlich ein Typ-Property Stundensatz beinhalten. Die Klasse soll auch ein lazy stored Property „Gehalt“ haben. Das Gehalt wird als  $\text{Stundenzahl} * \text{MeinStundensatz}$  berechnet. Dabei gilt:

$$\text{MeinStundensatz} = \text{Stundensatz} + (\text{Azahl der Jahre in der Firma}) * 0,75\text{€}$$

### Teilaufgabe 3

Eine Figur besteht aus zwei Rechtecken: rE1 und rE2. Die Position des rE1 darf durch setter geändert werden. Dementsprechend muss die Position von rE2 automatisch korrigiert werden, damit die Form der Figur bleibt. Realisieren Sie diese Funktionalität. Tip: Sie können Computed Properties bzw. Property Observer oder etwas Anderes, entsprechend Ihren Ideen benutzen.

## Teilaufgabe 4

Das Protokoll „log“ beinhaltet eine Funktion „logMe“, die als Parameter ein String hat. Diesen String gibt sie einfach in einer Form aus.

Die Klasse (oder Struktur) „Loggerin“ hat ein Property „delegate“, das das Protokoll „log“ Realisiert. Die Funktion „loge“ der Loggerin, die ein String als Parameter hat, ruft einfach die Methode „logMe“ des delegate auf.

Realisieren Sie 3 unterschiedliche Klassen (Strukturen) namens „Delegatin1“, „Delegatin2“ und „Delegatin3“, die das Protokoll „log“ realisieren. Die „logMe“ von DelegatinX gibt den Stringparameter aus, wenn dieser die Zahl X nicht beinhaltet. Sonst logt sie nichts. Testen Sie die drei Methoden, in dem Sie das delegate von der Loggerin entsprechend auf 3 unterschiedliche Delegatinnen setzen und die Funktion „loge“ aufrufen.