Spielzeuggeschäft

**Lerninhalte**: OO-Design, Vererbung, Superklasse, Schnittstellen, abstrakte Klassen, Polymorphismus, JUnit, JSON.

## Beschreibung

Aufgabe ist es eine Spielzeug Software für EOS zu entwickeln. Die EOS-Gruppe hat sich darauf spezialisiert Spielzeug Fahrzeuge aller Art zu vertreiben.

Die Aufgabe teilt sich in zwei große Teile auf, im ersten Teil soll erstmal eine Spielzeugverwaltung-Software entwickelt werden und im zweiten Teil wird die Software um die Lagerverwaltung erweitert

### Teil 1a)

In der Software soll es möglich seine verschiedenen Spielzeuge einzugeben, zu speichern (im JSON Format und lokal), zu editieren oder zu löschen. Jedes Spielzeug erhält bei der erstmaligen Eingabe eine generierte und einmalige Id. Zusätzlich muss geprüft werden, ob dieses Spielzeug in unserem Verwaltungssystem schon vorhanden ist und dem Benutzer darauf hinweisen und die Möglichkeit geben es zu editieren.

Spielzeuge:

Fahrrad, Motorrad, Auto, Lkw, Bulldozer, Segelboot, Motorboot, U-Boot, Luftkissenboot, Hubschrauber, Jet, Segelflieger.

Eigenschaften:

Id

Name

Farbe

Größe (M, L, XL)

Hersteller

Einkaufspreis

Verkaufspreis

Antriebsart

Anzahl der Räder

Datum

Lagerplatz (nur für Aufgabe Teil 2 wichtig)

Methoden:

Getter und Setter (außer Id)

Print()

Fahren() (Terminal Ausgabe -> Ich bin ein Auto und ich fahre auf dem Untergrund!)

Schwimmen()(Terminal Ausgabe -> Ich bin ein Motorboot und ich fahre auf dem Wasser!)

Fliegen()(Terminal Ausgabe -> Ich bin ein Jet und ich fliege in der Luft!)

lagerortAusgabe()(erst mit Teil 2 umsetzen!)

### Teil 1b)

Zusätzlich soll jetzt die Software um eine Suche erweitert werde. Mitarbeiter kann nach einem bestimmten Spielzeug suchen.

### Teil 2)

Die vorhandene Software sollte um ein weiteres Software Modul ergänzt werden und zwar der Lagerverwaltung. In der Lagerverwaltung ist es möglich die Spielzeuge, die angelegt wurden oder werden, einem Lagerplatz zuzuordnen und diesen bei Bedarf auszugeben.

Abgabe:

UML Model

Code

Wichtig:

-Alle public Methoden wurden mindestens mit einem Unit Test sinnvoll getestet!