Container füllen

Erarbeitung
der einfachen Version von
fuelle

Container füllen

- Schrittweise Erarbeitung nach dem Konzept des test-driven-development
- Formulierung (hier eines) Ziels

```
(fuelle
  '(30 30 30 30 20 20 20 20)
  '(80)
)
```

stuecke

container mit groesse

→ '(80 30 30 20)

Container füllen

- Schritt 1
  - Kopf der Funktion
  - Kennzeichnung der Unvollständigkeit

```
(define (fuelle stuecke container)
  'undefiniert)
```

Container füllen

- Schritt 1
  - Testen

```
(fuelle
  '(30 30 30 30 20 20 20 20)
  '(80)
)

  →  'undefiniert)
```

Container füllen

- Schritt 2
  - cond einbauen
  - erste Bedingung realisieren:
    Gibt es noch stuecke?

```
(define (fuelle stuecke container)
  (cond
    ((null? stuecke)
     #f)
    (else
     'undefiniert)
    ))
```

Container füllen

- Schritt 2
  - Testen

```
(fuelle '() *container*)
(fuelle
  '(30 30 30 30 20 20 20 20)
  '(80)
)
  → #f
  →  'undefiniert)
```

Container füllen

- Schritt 3
  - container exakt voll?
    (Erfolgsfall)

```
(define (fuelle stuecke container)
  (cond
    ((exakt-voll? container)
     container)
    ((null? stuecke) #f)
    (else  'undefiniert)
  ))
```

Container füllen

- Schritt 3
  - Testen
    (wird erst erfolgreich, wenn die Hilfsfunktion erfolgreich entwickelt wurde)

```
...
(fuelle
  '( <beliebig> )
  '(80 30 30 20)
)
  → '(80 30 30 20)
```

Container füllen

- Daher Schritt 4
  - Entwickeln Sie nach dem gleichen Prinzip die Hilfsfunktion *exakt voll?*
- Überlegen und bearbeiten Sie weitere Schritte nach dem gleichen Prinzip für die Funktion und gegebenenfalls notwendige Hilfsfunktionen