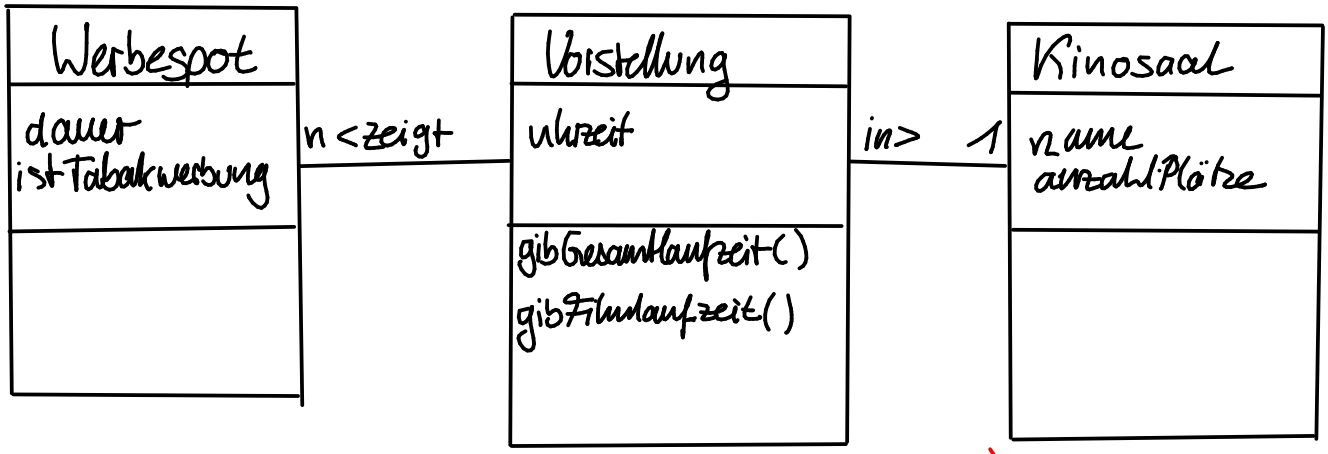
Musterlösung

Aufgabe 1a



(Vergessene Attr./Meth : -½ / Syntaxfehler je -½ / -1)

Aufgabe 1b

Mögliche Gründe:

- Anzahl Werbespots unbekannt / bei jeder Vorst. unterschiedlich
- Einfache Abarbeitung durch Ablauf in WarteSchlangen-Form
- Einfachere Algorithmen zum Einfügen

Anm: 2 Gründe hinreichend für volle BE! (je ½)

Aufgabe 2a

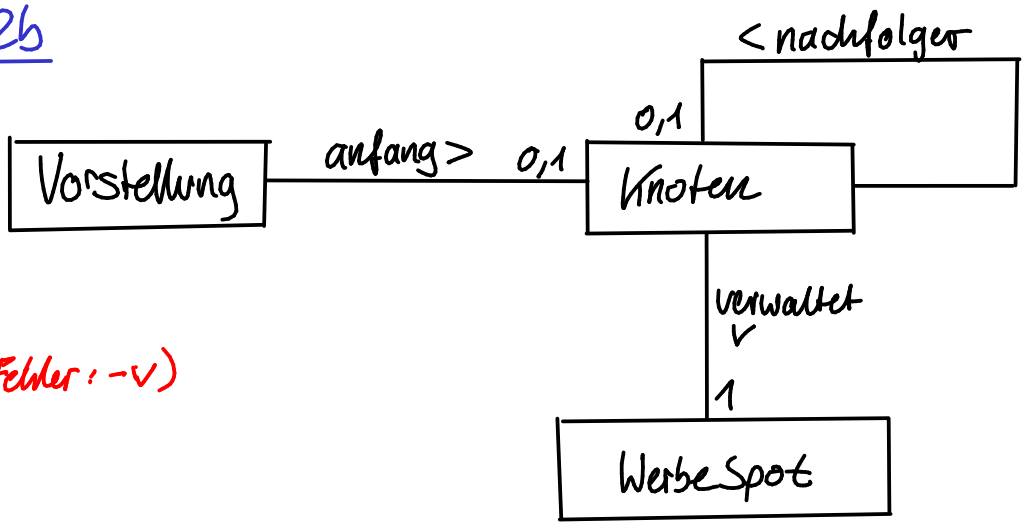
Da jedem Objekt der Klasse Werbespot der Nachfolger festgelegt wird, würde bei der zweiten Einordnung wieder der selbe Nachfolger gesetzt sein. Ergo würde sich in der ansonsten linearen Liste ein Zyklus entwickeln:

→ Vermeidung durch Trennung von Struktur und Inhalt!

→ 2b

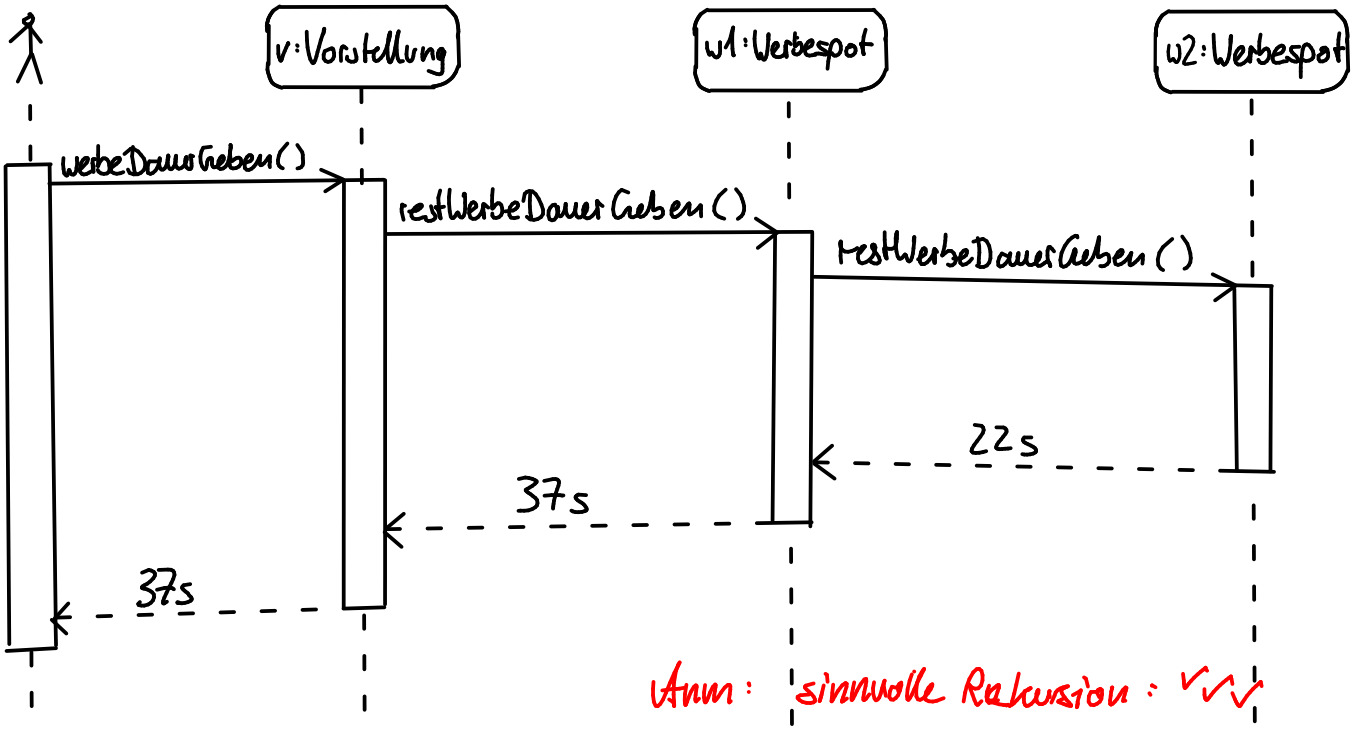
BE anteilig nach Grad der Erfassung / Beschreibung

Aufgabe 2b



(Je Fehler: -v)

Aufgabe 3a



Anm: sinnvolle Rekursion: vvv

Rest/Formalia: vvvv

Aufgabe 3b

In Klasse Vorstellung:

Methode werbeDauerGeben():

falls anfang existiert:

anfang.restDauerGeben() zurückgeben

sonst:

0 zurückgeben

} vvv

In Klasse WerbeSpot:

Methode restDauerGeben()

falls nachfolger existiert:

(dauerGeben() + nachfolger.restDauerGeben()) zurückgeben

sonst:

dauerGeben() zurückgeben

vv

Aufgabe 3c

(mögliche, hinreichende Begründung:)

Sofern nicht anders gespeichert, muss für die Gesamtwerbedauer t_{ges} und die Anzahl n der Werbespots je ein rekursiver Durchlauf erfolgen. Die durchschnittliche Zeit berechnet sich dann durch

$$\bar{t} = \frac{t_{ges}}{n}$$

(je nach Schlüssigkeit d. Antwort bis vvv)

Anm: Alternative Lösungen denkbar: (Widerlegungen)

- Speicherung von Anzahl od. GesamtDauer in Instanzvariablen der Vorstellung
- Ein komplexeres Datenobjekt {int; double} / z.B. als Obj. einer Klasse „Summe" wird als Rückgabewert einer rekursiven Methode „anzahlUndDauerGeben()" genutzt.
- Übergabe der beiden Werte zum durchzählen und aufsummieren als Parameter. Im „letzten" Listenobj. den Durchschnitt berechnen und unverändert rekursiv durchgeben.

Aufgabe 4a

Es handelt sich um das Entwurfsmuster Kompositum (Composite) ✓

vv Entwurfsmuster helfen dabei, für wiederkehrende Probleme (der Softwareentwicklung) Lösungsansätze bereit zu halten. Insbesonde bei der Arbeit in Teams können so allen vertraute Muster wiederverwendet werden.

Aufgabe 4b

Von abstrakten Klassen (hier: ListenElement) können keine Objekte erzeugt ✓ werden. Durch konkretere Subklassen werden die abstrakten Oberklassen weiter ausgestaltet ✓. Gemeinsame Bestandteile der Subklassen können jedoch bereits in der a. Klasse implementiert ✓ werden. (2 von 3 Aspekten hinreichend)

Aufgabe 4c

Die angegebene Idee führt nicht nur dazu, dass mehrere Playlisten aus Songs nacheinander ablaufen können, sondern ebenfalls in jeder der Playlisten rekursiv weitere Playlisten existieren können.

Technisch möglich könnte das zu unüberschaubaren Situationen für den Benutzer führen. Zudem sind sogar Zyklusbildungen möglich, die bei rekursiven Methodenaufrufen nur sehr schwer abzufangen sind.
Als konkrete Lösung wäre es einfacher und stabiler eine Klasse „Metaplaylist" zu erstellen, welche eine Liste von Playlisten enthält.

Anmerkung: Transferaufgabe zur Hinführung zu Bäumen und Graphen.

Anm: Viele Lösungen denkbar.
Wichtige Aspekte, die die Lösung beinhalten sollte:
- Es löst das gegebene Problem ✓✓
- es führt zu weiteren, komplexeren Problemen ✓✓
- Stellung bezogen oder sogar Vorschlag einer besseren Lösung: ✓