

## Analyse-Klassendiagramm: Finden von Operationen (zu MSt Operationssignaturen) → Balzert Kap. 2.4, 4.10, Siedersleben Kap. 3.4

**Operationskandidaten:** Verben im Text (Pflichtenheft, Anwendungsfallbeschreibung).

**Name:** Verb im Infinitiv. Aus Benutzersicht,

**Bsp.:** Statt buchVerleihen() besser buchAusleihen()!

**Signatur:** operation(in par1: Typ1 = Wert, out par2: Typ2): Ergebnistyp

**Bsp.:** leser.ausleihen(in buch: Buch): Datum

**Objektbezug-Klassifikation:** Nutzbar als Stereotyp im Operationen-Feld

◆ **Konstruktoroperation:** erzeugt und liefert ein neues Objekt der Klasse: «constructor»

◆ **Destruktoroperation:** vernichtet ein existierendes Objekt: «destructor»

◆ **Klassenoperation:** arbeitet auf mehreren Objekten der Klasse, **Bsp.:** suchen(), summieren()

◆ **Objektoperation:** arbeitet auf existierendem Objekt: ohne Kennzeichnung

- **abstrakte Objektoperation:** Objektoperation, die in Unterklasse definiert werden muss: *kursiv*

**Aufgabe 1:** Mitarbeiter-Operationen einordnen

**Datenfluss-Klassifikation:** Nutzbar als Stereotyp im Operationen-Feld

◆ **Kommando:** Ändert den Systemzustand, **Bsp.** in Klasse Mitarbeiter:

«command»

gewinnbeteiligungenFestsetzen(gesamtsumme: Geldbetrag)

gehaltAnpassen(faktor: float)

dienstreiseGenehmigen(...)

versetzenNach(:Abteilung)

◆ **Abfrage:** Nur Lesezugriff/Berechnung ohne Änderung des Systemzustands, **Bsp.:**

«query»

nachWohnortSuchen(plz: PLZ): List<Mitarbeiter>

stundenlohnBerechnen(): Geldbetrag

ausweisDrucken()

**Reihenfolge:** 1. Konstruktor-, 2. Destruktor-, 3. Kommandos a) Klassen-, b) Objekt-, 4. Abfragen dito

**Aufgabe 2:** Student modellieren mit Klassifikation-Stereotypen der Operationen in dieser Reihenfolge

**Sichtbarkeit-Klassifikation:**

◆ **extern:** Wird von außerhalb des Systems, z. B. vom UI aktiviert, prüft Geschäftsregeln. In Analyse!

◆ **intern:** Wird nur von anderen Operationen aktiviert. Erst im Entwurf ermitteln.

**Objektverwaltungsoperationen:** Immer notwendig

◆ **interne:** Elementar, ungeprüft, für jedes Objekt nötig. Erst im Entwurf!

- new(): Konstruktoroperation: Erzeugen eines neuen Objekts

- delete(): Destruktoroperation: Vernichten eines bestehenden Objekts

- set/get-Attribute(): Ändern eines Attributwerts, **Bsp.:** setMonatsgehalt()

- link(): Aufbauen einer Verbindung zwischen Objekten (Verbindungsverwaltung)

- unlink(): Abbauen einer Verbindung zwischen Objekten (Verbindungsverwaltung)

- getlink(): Liefern des Partners einer Verbindung zwischen Objekten (Verbindungsverwaltung)

◆ **externe:** Mit Prüfungen und Folgeoperationen. Wenn sehr modellauflähdend, erst im Entwurf!

- erfassen(): Konstruktoroperation: Erzeugen eines neuen Objekts

- ändern(): Ändern der Attributwerte eines bestehenden Objekts

- löschen(): Destruktoroperation: Vernichten eines bestehenden Objekts

- suchen(...): Alle/bestimmte(s) Objekt(e) der Klasse liefern [Balzert: erstelleListe]. Varianten finden!

**Aufgabe 3:** Modellieren Sie: Ein Kunde kann mehrere Konten bei einer Bank eröffnen und von einem Konto auf ein anderes überweisen. In welche Klasse „Konto eröffnen“ respektive „überweisen“?

**Aufgabe 4:** Modellieren Sie: Ein Leser kann mehrere Bücher mit Leihfrist ausleihen. In welche Klasse „Buch ausleihen“?