

Belegsatz

Fachinformatiker Anwendungsentwicklung (AO 1997)
Fachinformatikerin Anwendungsentwicklung (AO 1997)
1196

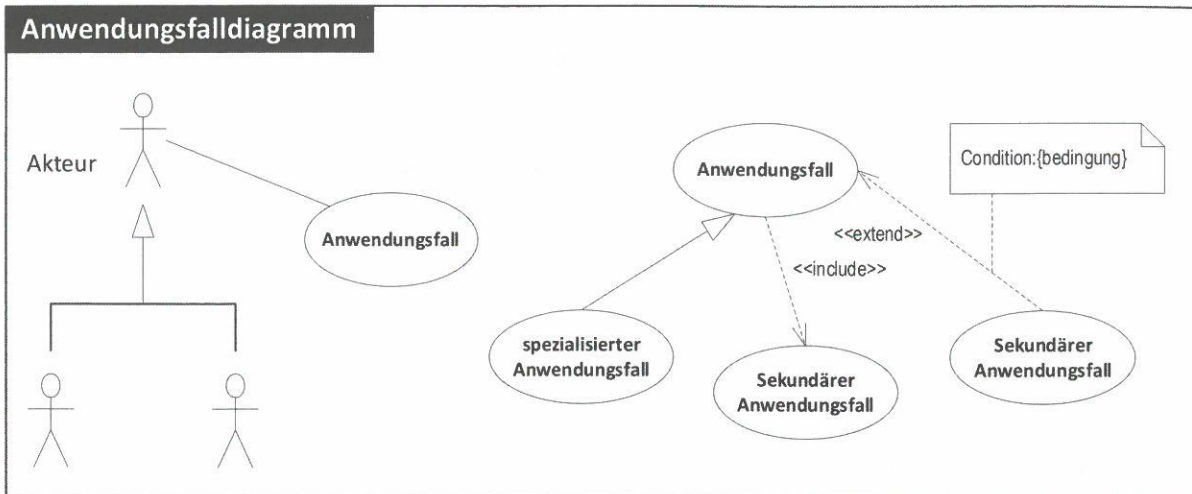
1

Ganzheitliche Aufgabe I Fachqualifikationen



Inhalt

UML-Anwendungsfalldiagramm	2
UML-Zustandsdiagramm	2
UML-Sequenzdiagramm	3
UML-Klassendiagramm	4
Pseudocode	4
SQL-Syntax (Auszug)	5/6

UML-Anwendungsfalldiagramm



UML-Zustandsdiagramm, Notation (Auszug)

 <div data-bbox="201 851 367 918"> <p>Zustand</p> </div>	<p>Anfangszustand</p> <p>Zustand, den ein Element einnehmen kann. z. B. Element = Fenster; Zustände: offen, geschlossen</p>
<p>Ereignis[Bedingung]/Handlung →</p> <p>Beispiel</p> <div data-bbox="121 1034 446 1088"> <p>offen → /schließen → zu</p> </div>	<p>Zustandsübergang (Transition) von einem Quellzustand zu einem Zielzustand. In der Beschriftung kann Folgendes angegeben werden:</p> <ul style="list-style-type: none"> - Ein Ereignis (trigger), welches den Zustandsübergang auslöst - Eine Bedingung (guard), welche beim Zustandsübergang erfüllt sein muss - Eine Handlung (action), welche den Zustandsübergang bewirkt
<div data-bbox="140 1120 343 1216"> <p>Zustand</p> </div> <p>Ereignis[Bedingung]/Handlung</p>	<p>Selbstaufzuruf</p>
	<p>Endzustand</p>

Sequenzdiagramm

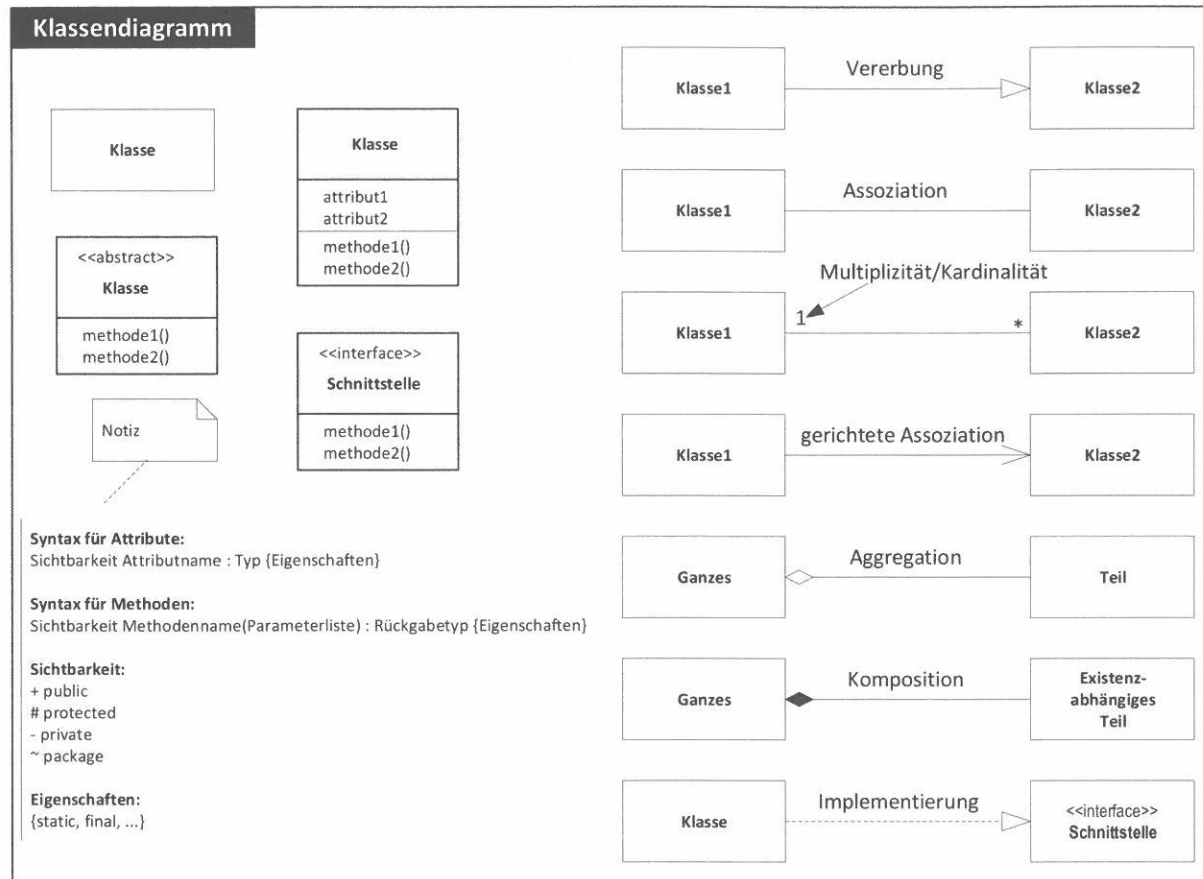
```
sequenceDiagram
    participant Sender
    participant x as x:Klasse1
    participant y as y:Klasse2
    participant xX as :KlasseX

    Sender->>x: Nachricht()
    alt [i>0]
        x->>y: nachricht1()
        y-->>x: 17
    else [else]
        x->>y: nachricht2()
    end
    x->>x: Selbstaufuf
    x-->>x: Ausführung(opt.)
    x-->>x: 
    x--x: 
    x--x: X
    x--x: Objekt-Destruktion
    x-->>xX: create
    xX-->>x: Objekt-Konstruktion
```

synchrone Nachricht:
Der Sender erwartet vom Empfänger ein Beenden des Aufrufs, Rückgabewert (z.B. 17) möglich

asynchrone Nachricht:
Der Sender erwartet vom Empfänger keine Antwort

UML-Klassendiagramm



Pseudocode

Pseudocode	Beschreibung
wenn [Bedingung] dann [Anweisung] sonst [Anweisung] Ende wenn	Verzweigung
zähle [Variable] von [Startwert] bis [Endwert] [Schleifeninhalt] Ende zähle	Zählschleife (Iteration)
solange [Bedingung] [Schleifeninhalt] Ende solange	Kopfgesteuerte Schleife
wiederhole [Schleifeninhalt] solange < Bedingung >	Fußgesteuerte Schleife
:=	Zuweisung
=; <; >; <=; >=; < >	Vergleiche
integer	
double	
string	
array	

SQL-Syntax (Auszug)

Syntax	Beschreibung
<i>Tabelle</i>	
CREATE TABLE Tabellennamen(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellennamen ADD COLUMN Spaltenname Datentyp DROP COLUMN Spaltenname Datentyp ADD FOREIGN KEY (Spaltenname) REFERENCES Tabellennamen(Primärschlüsselspaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) REFERENCES Tabellennamen(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellennamen	Löscht eine Tabelle
<i>Befehle, Klauseln, Attribute</i>	
SELECT * Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... (SELECT ... FROM ... WHERE ...) AS xyz FROM ... WHERE ...	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: xyz) ausgegeben.
SELECT DISTINCT	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery) WHERE NOT EXISTS (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXISTS negiert die Bedingung.
GROUP BY Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [,Spaltenname2, ...] ASC DESC	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
<i>Syntax Beschreibung</i>	
<i>Datenmanipulation</i>	
DELETE FROM Tabellennamen	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellennamen SET	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellennamen VALUES (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder SELECT ... FROM ... WHERE	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind

Fortsetzung SQL-Syntax →

Aggregatfunktionen	
AVG (Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT (Spaltenname *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM (Spaltenname Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN (Spaltenname Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT (Zeichenkette, Anzahlzeichen)	Liefert <i>Anzahlzeichen</i> der Zeichenkette von links.
RIGHT (Zeichenkette, Anzahlzeichen)	Liefert <i>Anzahlzeichen</i> der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT (time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE (Wert)	Wandelt einen Wert in ein Datum um
DAY (Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH (Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY (Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR (Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD (Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF (Datumsteil, Anfangsdatum, Enddatum) Datumsteile: DAY, MONTH, YEAR	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Textattributen auf Gleichheit, Verwendung von Platzhaltern möglich.
NOT	Logische Negation
OR	Logisches ODER
IS	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2018-03-29