Verbundstudiengang Wirtschaftsinformatik (B.Sc.)
Algorithmen und Programmierung II
Übungsaufgaben

Prof. Dr. Anja Haake
Veranstaltung am 13.04.2024 (T3)

Aufgabe 1 – Zusicherungen als boolesche Ausdrücke schreiben können

Formulieren Sie folgende Zusicherungen als boolesche Ausdrücke und Zusicherungen in Java:

(a) Eine Zahl x ist negativ.
(b) Eine ganze Zahl y ist durch 2 teilbar.
(c) Eine ganze Zahl z ist nicht negativ und durch 3 teilbar.

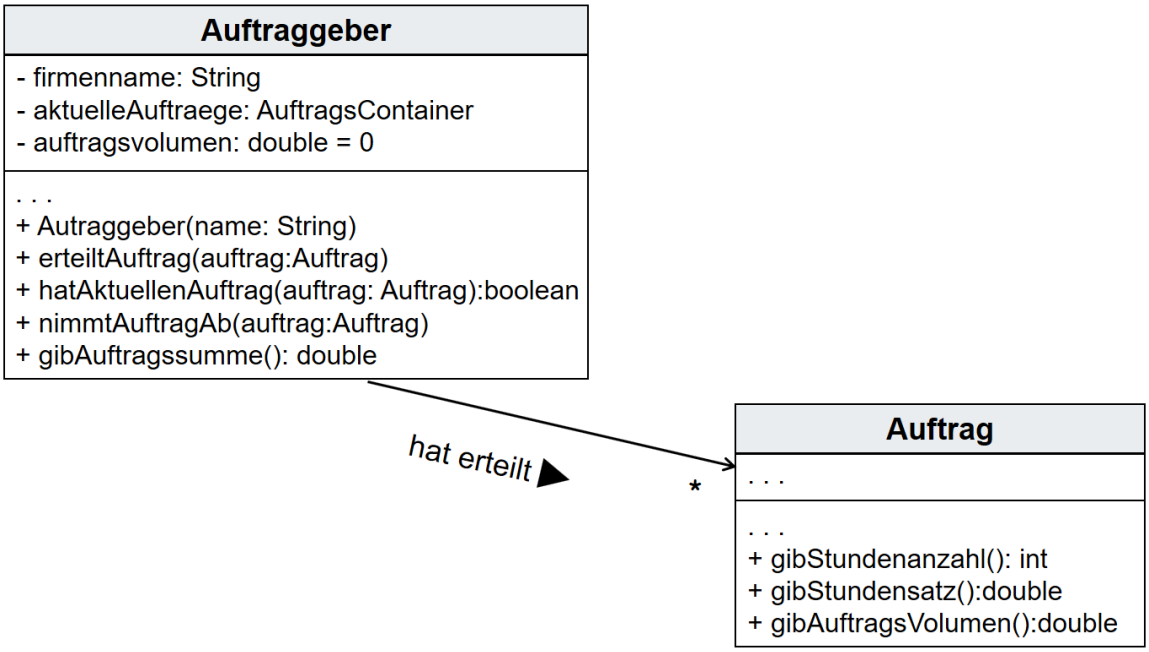
Aufgabe 2 – Einfache Programme mit Vor- und Nachbedingungen spezifizieren

Spezifizieren Sie die Vor- und Nachbedingungen für folgende Programme:

(a) Ein Programm berechnet zu zwei Zahlen X und Y den Mittelwert m = (X+Y)/2.
(b) Ein Programm berechnet zu zwei Zahlen X und Y den Mittelwert m = (X+Y)/2, x und y bleiben dabei unverändert.

Aufgabe 3 – Funktionstests für objektorientierte Programme mittels Formulierung von Vor- und Nachbedingungen als UnitTests

Gegeben sei folgendes UML-Klassendiagramm…



… und (s. nächste Seite) …

…folgender Rumpf einer JUnit-Testmethode (vgl. LE 6, Kap. 2.6.3, S. 61 ff):

```
01 Auftraggeber auftraggeber1;
02 Auftrag auftrag1;
03 auftraggeber1 = new Auftraggeber("Sonnen Immobilien GmbH");
04 auftrag1 = new Auftrag(2, "Beratung", 10, 125, "18.01.04");
05 auftraggeber1.erteiltAuftrag(auftrag1);
06 assertTrue(auftraggeber1.hatAktuellenAuftrag(auftrag1));
07 double ursprünglicheAuftragssumme = auftraggeber1.gibAuftragssumme();
08 double abgearbeiteteSumme = auftrag1.gibStundensatz()
                              * auftrag1.gibStundenanzahl();
09 auftraggeber1.nimmtAuftragAb(auftrag1);
10 assertFalse(auftraggeber1.hatAktuellenAuftrag(auftrag1));
11 assertEquals("Auftragsabnahme",
                ursprünglicheAuftragssumme - abgearbeiteteSumme,
                auftraggeber1.gibAuftragssumme());
```

(a) Identifizieren Sie in dem gegebenen Rumpf die Anteile
- Herstellung der Exemplare im gewünschten Zustand
- Vorbedingung
- Erhebung von Sollwerten (Erwartungswerten)
- Aufruf der zu testenden Methode
- Nachbedingung.

(b) Welche Funktionalität wird durch die Testmethode getestet?

(c) Welchen Bezeichner sollte die Testmethode tragen?

(d) Konzipieren Sie einen Funktionstest für den Konstruktor der Klasse Auftrag.
Formulieren Sie den Test als JUnit-Test.

JUnit stellt zum Prüfen von Zusicherungen u.a. folgende Methoden bereit:
- `assertNull(Object o)` überprüft, ob die Referenz o gleich `null` ist.
- `assertNotNull(Object o)` überprüft, ob die Referenz o ungleich `null` ist.
- `assertEquals(int x, int y)` überprüft die Gleichheit zweier `int`-Werte.
- `assertEquals(double x, double y)` überprüft die Gleichheit zweier `double`-Werte.
- `assertEquals(String x, String y)` überprüft die Gleichheit zweier Zeichenketten.