

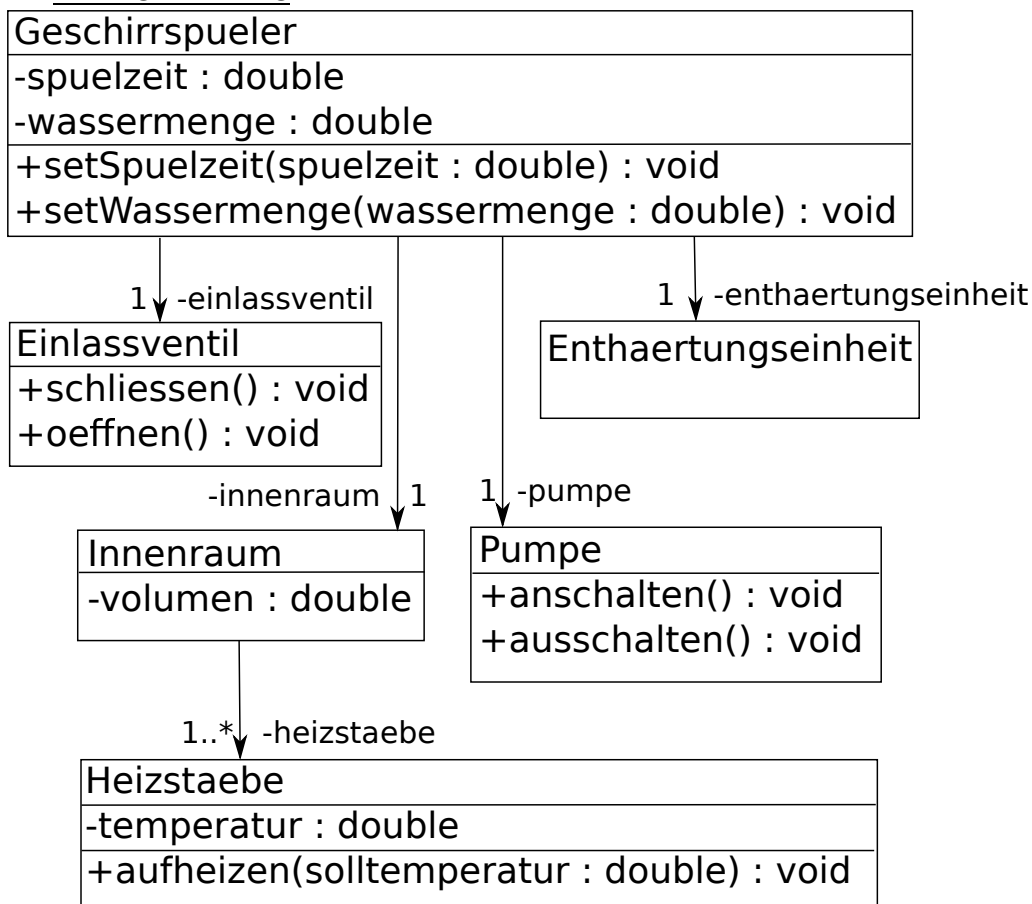
7. Übungsblatt - Informatik 1 - Lösungsbeispiele

Aufgabe 1 (UML Klassendiagramm)

Beschreiben Sie folgenden groben Aufbau einer Geschirrspülmaschine als vollständiges UML-Klassendiagramm. Beachten Sie die Java-Namenskonventionen.

Eine Geschirrspülmaschine besteht aus einem Einlassventil, welches sich schließen oder öffnen kann; aus einer Enthärtungseinheit; dem Innenraum mit Heizstäben; sowie einer Pumpe. Der Innenraum hat ein Volumen in Kubikmeter. Die Heizstäbe heizen das Wasser auf eine vorgegebene Solltemperatur auf. Die Pumpe kann an- oder ausgeschaltet werden. Die Menge des einzulassenden Wassers (in Litern) und die Spülzeit (in Minuten), die ein Spülvorgang benötigt, lässt sich bei der Geschirrspülmaschine einstellen.

Lösungsvorschlag:

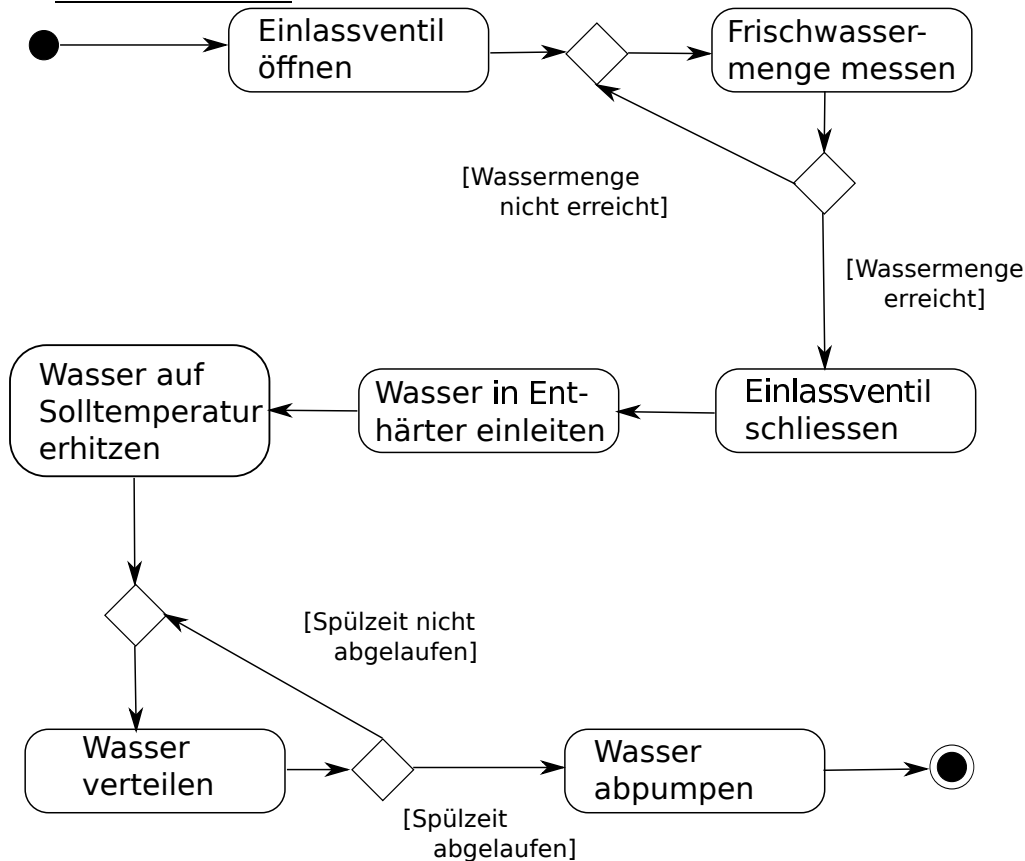


Aufgabe 2 (UML Aktivitätsdiagramm)

Der folgende Ablauf ist eine vereinfachte Beschreibung eines Spülvorgangs einer Geschirrspülmaschine. Geben Sie diesen Ablauf als UML-Aktivitätsdiagramm an. Verwenden Sie keine UML Abkürzungen.

Zu Beginn wird das Einlassventil geöffnet. Danach wird die Menge des eingelassenen Frischwasser gemessen, bis die Menge des einzulassenden Wassers erreicht ist. Dann schließt das Einlassventil. Das Wasser wird über die Enthärtungseinheit geleitet und gelangt dadurch in den Innenraum. Dort wird das Wasser auf die eingestellte Solltemperatur erhitzt. Danach wird das Wasser von der Pumpe im Innenraum verteilt, bis die Spülzeit abgelaufen ist. Zum Schluss wird das Wasser abgepumpt.

Lösungsvorschlag:



Aufgabe 3 (Heap, Stack)

Betrachten Sie folgende Klasse Person und zugehörige main-Methode. Beschreiben sie grafischen den Zustand des dynamischen Speicherbereichs (Heap) und des Laufzeitkellers (Stack) beim letzten Aufruf der Objektmethode laufen. Der letzte Methodenaufruf von laufen soll dabei noch nicht beendet, aber in der letzten Zeile der Methode angelangt sein.

```

public class Person {
    private String nachname;
    private int gelaufeneKilometer;

    public void main(String [] args) {
        Person albert = new Person();
    }
}
  
```

```

    albert.nachname = "Albert";
    albert.gelaufeneKilometer = 11;
    Person anna = new Person();
    anna.nachname = "Anna";
    albert.laufen();
    albert.laufen();
    anna.laufen();
}

public void laufen() {
    boolean muede = this.gelaufeneKilometer > 20;
    if (muede) {
        this.gelaufeneKilometer += 5;
    } else {
        this.gelaufeneKilometer += 10;
    }
}
}

```

Lösungsvorschlag:

