

Nennen Sie die zwei Kategorien, in die Operationen eingeteilt werden, die **keine** Erzeugungsprozeduren (Konstruktoren) sind.

- Kommando
- Abfrage

Wie viele Argumente hat eine Operation in gut geschriebener OO-Software?

- Meistens eines oder keines

Welche der folgenden Aussagen über Abfragen sind richtig?

- Sie sind idempotent
- Sie dürfen den Zustand eines Objekts nicht ändern

Welche der folgenden Aussagen über Kommandos sind richtig?

- Sie ändern den Zustand des Objektes auf dem sie aufgerufen wurden
- Sie können den Zustand anderer Objekte verändern

Welche Folgen hat es, Kommandos und Abfragen im Code nicht strikt voneinander zu trennen?

- Die Software ist schwer verständlich
- Die Software ist schlechter wartbar

Wählen Sie die korrekte Aussage über Ausdrücke aus.

- Steht für einen Wert, der von einer Instruktion benutzt wird

Wählen Sie alle richtigen Aussagen über Instruktionen aus

- Sie sind elementare Operationen
- Sie werden während der Programmausführung ausgeführt

Ordnen Sie die Begriffe Syntax, Semantik und Pragmatik den richtigen Beschreibungen zu.

- Gültige Strukturen von Sätzen (Syntax)
- Gültigkeit wird vollständig vom Compiler geprüft (Syntax)
- Bedeutung gültiger Sätze (Semantik)
- Erwarteter Effekt eines Programms (Semantik)
- Auswirkung gültiger Sätze (Pragmatik)
- Auswirkung des Programms auf die reale Welt (Pragmatik)

Ordnen Sie die folgenden Regeln von präzise zu abstrakt

- Semantische Regeln
- Statische Semantikregeln
- Syntaktische Regeln
- Lexikalische Regeln

Ordnen Sie den folgenden Ausdrücken ihren Typ zu

- `paule.getLocation()` (Location)
- `territory.getDefaultHamster()` (Hamster)
- „Paule ist cool“ (String)
- $6 * 5 / 2$ (Integer)
- `paule.frontIsClear()` (Boolean)
- `Territory.isFree(new Location(2,3))` (Boolean)

Ist ein Objekt A durch die Klasse B definiert, dann

- Nennt man B die erzeugende Klasse von A
- Nennt man A die Instanz von B

Nennen Sie die drei Vertragsarten für Klassen und Operationen

- Vorbedingungen
- Nachbedingungen
- Klasseninvarianten

Wann muss eine Operationsimplementierung sicher stellen, dass die Nachbedingung am Ende ihrer Operationsausführung erfüllt ist?

- Wenn die Vorbedingung zu Beginn der Ausführung erfüllt ist

Vor-, Nachbedingungen & Klasseninvarianten

- Klasseninvarianten oder Klasseninvariante sind Bedingungen, die zwischen beliebigen Operationsausführungen auf den Objekten der Klasse gelten müssen. Eine Nachbedingung ist eine Bedingung, die einer Operation für eine korrekte Realisierung durch den Anbieter auferlegt ist. Dahingegen ist eine Vorbedingung eine Bedingung, die eine Operation jedem ihrer Aufrufer für einen erfolgreichen Aufruf auferlegt.

Wann muss ein Ausrufer beim Vorbedingungsprinzip sicherstellen, das die Vorbedingung erfüllt ist?

- Vor dem Aufruf der Operation

Gegeben sei der oben abgebildete Code. Ordnen Sie jeder der folgenden Beschreibungen zu, ob es sich um eine Klasseninvariante oder eine Vor- oder Nachbedingung für die Operation `pickGrain()` handelt.

```
class Hamster {
    /* ... */
    void pickGrain() {
        // Hamster hebt ein Korn auf
    }
}
```

- Der Hamster ist initialisiert (Klasseninvariante)
- Auf der Kachel unter dem Hamster liegen mindestens ein Korn (Vorbedingung)
- Der Hamster hat ein Korn mehr im Mund (Nachbedingung)
- Auf der Kachel unter dem Hamster liegt ein Korn weniger (Nachbedingung)
- Der Hamster hat 0 oder mehr Körner im Mund (Klasseninvariante)

Welche der folgenden Eigenschaften ist nicht Teil des minimalen Qualitätsstandards für Kommentare?

- Erklärung des exakten Implementierungsdetails