

PSE Blatt 04 - Gruppe 32

Marvin Hepp (3759031)& Nick Daiber (3728224)

November 22, 2024

1

a

V ist wahr, wenn Paule den Sonnenaufgang beobachten kann impliziert, dass er nach Osten Blickt und nicht vor einer Wand steht

W ist wahr, wenn Paule nach Osten blickt und nicht vor einer wand steht oder nicht den Sonnenaufgang beobachtet (Vereinfacht)

W ist wahr, wenn nicht Paule nicht nach Osten blickt oder vor einer wand steht oder nicht den Sonnenaufgang beobachtet (unvereinfacht, aber echt nicht schön)

b

X	Y	$\neg X$	$\neg Y$	$\neg X \vee Y$	$\neg(\neg X \vee Y)$	$X \wedge \neg Y$
0	0	1	1	1	0	0
0	1	1	0	1	0	0
1	0	0	1	0	1	1
1	1	0	0	1	0	0

c

$(Z) \ || \ (Z \ \&\& \ (X \ \&\& \ (!Y)))$

Mit Funktionen eingesetzt

```
(!paule.canViewSunrise()) ||  
(paule.canViewSunrise() &&  
((paule.getDirection() == Direction.EAST) && (!paule.frontIsClear()))))
```

2

a

Strikte Operatoren werten jeden Teil des Ausdrucks aus, Semistrikte brechen schon zuvor ab bsp

$(x \neq 0) \ \&\& \ 2/x > 1$

Führt bei Strikter Auswertung zum Fehler, bei Semistrikter nicht

b

i

- E_1 kann immer Ausgewertet werden, da sowohl `ham` als auch `null` definiert sind und miteinander verglichen werden können
- E_2 führt zu einem Fehler, da `null` keine Methode `grainAvailable` hat.
- E_3 führt zu einem Fehler, da `null` keine Methode `grainAvailable` hat.

ii

- E_1 kann immer Ausgewertet werden, da sowohl `ham` als auch `null` definiert sind und miteinander verglichen werden können
- E_2 führt zu einem Fehler, da `null` keine Methode `grainAvailable` hat.
- E_3 kann Ausgewertet werden, da nach `ham == null (==True)` “abgebrochen” wird

b

E_2

3

a

Das Atribut `name` darf nicht `null` sein.

Die Referenz `hamster` darf nicht `numm` sein.

Der Wert von `grainNumber` muss eine ganzzahl \geq dem überziehungslimit sein.

Der Wert von `maximumOverdraft` muss eine ganzzahl ≤ 0 sein.

b

Wenn auf `secondaryHamster` zugegriffen wird, als sei es ein `Hamster`-Objekt kann es zu problemen kommen, besser wäre beispielsweise einen Vektor zu nutzen, besonders, da so mehr als nur 2 besitzer angegeben werden können.

c

Vorbedingungen

- Der Hamster muss mindestens so viele Körner bei sich haben, wie er einzahlen kann.
- Der Hamster muss an dem Automaten stehen

Nachbedingungen

- Der Kontostand muss um die eingezahlten Körner erhöht worden sein
- Der Hamster darf die eingezahlten Körner nicht mehr besitzen.

d

Vorbedingungen

- Der Hamster muss am Automat stehen
- Es müssen genug Körner auf dem Konto sein oder das überziehungslimit muss für das abheben genügen

Nachbedingungen

- Der Hamster muss die abgehobenen Körner bei sich haben
- Der Kontostand muss um die Anzahl der abgehobenen Körner reduziert worden sein

e

`this` ist eine Referenz auf das Objekt dem die methode angehört. `null` ist eine Referenz auf nichts

f

Zeile 9: Es wird 3 ausgegeben, da der Bezeichner `firstAccount` seit Zeile 7 auf das zuletzt erstellte Konto-objekt zeigt, welches mit 3 Körnern initialisiert wurde.

Zeile 10: es wird 35 ausgegeben, da der Bezeichner `secondAccount` auf das zuerst erstellte Kontoobjekt zeigt und dessen Kontostand in Zeile 8 von 5 auf 35 erhöht wurde (insofern 30 Körner aufgeladen werden konnten).

Zeile 11: Es wird nochmal 3 ausgegeben, da der Bezeichner `thirdAccount` seit beginn auf das zuletzt erstellte Objekt zeigt.

g

```
transferGrainsTo(firstAccount, null, 20)
```

Es wird versucht die `depositGrains` methode aufzurufen obwohl `tooBankAccount` auf `null` zeigt.

h

in offensichtlichen Fällen wie den oben genannten schon. Eine Referenz auf `null` kann aber auch erst zur laufzeit entstehen, wenn ein Objekt gelöscht wurde, eine Referenz auf dieses aber nicht.