Projektaufgabe 2

Jan-Erik Menzel, Steffen Anhäuser, Lukas Hofmann, Melina Morch

Anmerkungen:

- Wir planen letztendlich eine Konsole, auf der ein Kalender abgebildet wird und durch Eingaben werden Veranstaltungen geladen und getestet, ob es Terminkollisionen gibt
- Unserer Meinung nach ist Code trotz oder gerade durch Sonderzeichen sprechend und lesbar. Also sprechend in dem Sinne, dass nicht viel Kenntnisse vorhanden sein müssen, um zu verstehen, was die Codezeilen bewirken. Die Sonderzeichen bewirken klare Trennung und sichtbares Erkennen, wann ein Codeblock „zu Ende“ ist.
- Dann haben wir das nicht ganz richtig verstanden, uns war es wichtig, dass man sich die Typen der Variablen so anpassen kann wie man sie braucht, aber nicht dass diese sich nach der Initialisierung ändern.
- Trotz dem soll es für die Variablen einen Default-Case (also Daten oder Events) geben, der standardisiert immer eintritt, es sei denn es wird explizit anders gewünscht.
- Git-Like soll heißen, dass es mehrere Bearbeiter geben können soll, also unterschiedliche Anwender sollen auf gleiche Veranstaltungen auf der Konsole zugreifen können und je nach festgelegten Zugriffsrechten Veränderungen handhaben.

Abstrakte Grammatik

Syntax

t :==	Terms
true	constant true
false	constant false
if(x) {y} else {z}	conditional
for(i)	loop
x := y	declaration

operationale Semantik

t:== terms

true

false

if(x) {y} else {z}

for(i)

x := y

v:== values

true

false

int x

Event e

TimeWindow z

Date d

… usw.

(Datentypen im klassischen Sinne Teil der operationalen Semantik?)