PSp
11. 12. 2018

B

Zwischenprüfung
Teil Rege

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Name, Vorname	Klasse	Punkte	Note

Aufgabe 1: Smalltalk (3 Punkte)

Was ergeben die folgenden Smalltalk Ausdrücke?

```
1 + 5/4.
```
_____

```
5/4 + 1.
```
_____

```
2 + 3 * 2 squared.
```
_____

```
1 + 2 * 2 raisedTo: 2.
```
_____

```
1/4 + 5/4.
```
_____

```
[:x | 3 + x / 2] value: 7.
```
_____

Aufgabe 2: WASM Code (3 Punkte)

Gegeben sei folgende Funktionen double foo(double arg0, double arg1); die Methode verwendet eine zusätzliche lokale double Variable.

Der WASM Code sieht folgendermassen aus:

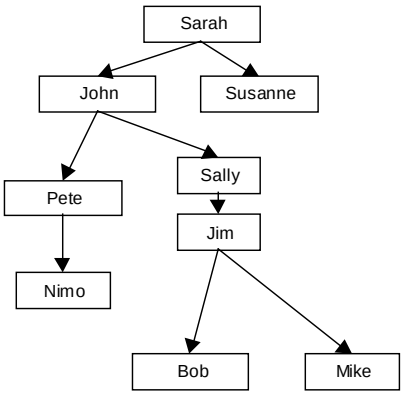
```
(func (;1;) (type 1) (param f64 f64) (result f64)
   (local f64)
   get_local 0
   get_local 1
   f64.div
   set_local 2
   get_local 2
   i32.nearest/f64
xxxx
   i32.const 1
   i32.eq
   if
     get_local 0
     f64.const 1.0
     f64.add
     return
   end
   get_local 0
   f64.const 2.0
   f64.sub
   return
```

Welcher Wert wird zurückgegeben wenn die Funktion mit foo(42,41) aufgerufen wird?

_____

Aufgabe 3: Prolog (6 Punkte)

Gegeben sei folgender Stammbaum



Als Prolog Regelbasis:

```
elternteil(sarah, john).
elternteil(john, sally).
elternteil(jim,  mike).
elternteil(sarah, susanne).
elternteil(sally, jim).
elternteil(jim, bob).

elternteil(john,pete).
elternteil(pete,nimo).
mannlichlist([john, jim, mike,bob,
pete, nimo]).
weiblichlist([sally, sarah,
susanne]).
```

a) Sie wollen herausfinden, ob eine Person eine Frau ist und dafür das entsprechende Prädikat: frau(X) definieren.

- `frau(X) :- member(X, weiblichlist (L).`
- `frau (X) :- X in weiblichlist (L).`
- `frau (X) :- mannlichlist(L),not(member(X,L)).`
- `frau (X) :- weiblichlist (X,L).`

b) Sie wollen den Mutter(V) einer Person(P) herausfinden.

- `mutter(V,P) :- weiblichlist (L),member(V,L).`
- `mutter (V,P):- elternteil(V,P), mannlichlist (L)`
- `mutter (V,P) :- X in weiblichlist (L).`
- `mutter (V,P) :- elternteil(V,P), weiblichlist (L),member(V,L).`

c) Sie sollen für Person (P) sämtliche Nachfahren (N) bestimmen

-
```
nachfahre(N,P) :- elternteil(N,P).
nachfahre(N,P) :- elternteil(N,Z),nachfahre(Z,P).
```

-
```
nachfahre (N,P) :- nachfahre(Z,P), elternteil(N,Z).
nachfahre(N,P) :- elternteil(N,P).
```

- `nachfahre(N,P) :- elternteil(N,P); nachfahre(N,P).`

```
nachfahre(N,P) :- elternteil(N,P).
nachfahre(N,P) :- elternteil(Z,N),nachfahre(P,Z).
```