# Switch-Case Anweisung über bool und int32

CPIB, HS2016 – Gruppe 10: Remo Rossi und Roman Meier

# Einleitung

## Switch-Case mit Beispiel

IML verfügt über keine Switch-Case Anweisung. Wenn auch diese über eine verschachtelte IF-Anweisung umsetzbar wäre, ist eine Switch-Case oft übersichtlicher und einfacher verständlich. Um die Funktionsweise und den Einsatz zu verdeutlichen ein Besipiel.

program SwitchCase(in n:int32, out f:int32, in b:bool, out r:int32)

global

fun switchIntFun(n:int32) returns var res:int32

do

res init := 0;

switch n

case 1 : res := n + n;

res := n - n

case 2 : res := n \* n

case 3 : res := n \* n + n

casedefault :

endswitch

endfun;

do

f init := switchInt(n)

r init := swicthBool(r)

endprogram

## Lexikalische Erweiterung

Lexikalisch wird die Syntax um folgende Token erweitert:

|  |  |
| --- | --- |
| Pattern | Token |
| switch | SWITCHINT32 |
| endswitch | ENDSWITCH |
| case | (CASE, TYPE) |
| casedefault | CASEDEFAULT |

## Grammatikalische Erweiterung

Cmd ::= SKIP

| expr BECOMES expr

| IF expr THEN cpsCmd ELSE cpsCmd ENDIF

| WHILE expr DO cpsCmd ENDWHILE

| CALL IDENT exprList [globInits]

| DEBUGIN expr

| DEBUGOUT expr

| SWITCH IDENT //typeident

CASE LITERAL COLON cpsCmd

[{CASE LITERAL COLON cpsCmd}]

CASEDEFAULT COLON cpsCmd

ENDSWITCH