Programmierung WS 18
Hausaufgaben - Blatt 9

Julian Giesen (MNR 388487)
Levin Gäher (MNR 395035)
Gruppe 12

HA 2

```
-- Types
data VariableName = X | Y deriving Show
data Expression = Constant Int | Variable VariableName | Add Expression Expression |
    Multiply Expression Expression deriving Show

-- Resolves Variables
getValue :: VariableName -> Int
getValue X = 5
getValue Y = 13

-- Resolves Expressions
evaluate :: Expression -> Int
evaluate (Constant c) = c
evaluate (Variable v) = getValue v
evaluate (Add ex1 ex2) = (evaluate ex1) + (evaluate ex2)
evaluate (Multiply ex1 ex2) = (evaluate ex1) * (evaluate ex2)

-- Optimizes Expressions One Level
tryOptimize :: Expression -> Expression
tryOptimize (Add (Constant c1) (Constant c2)) = Constant (c1 + c2)
tryOptimize (Multiply (Constant c1) (Constant c2)) = Constant (c1 * c2)
tryOptimize ex = ex

-- Optimizes Expressions Recursively
evaluatePartially :: Expression -> Expression
evaluatePartially (Add ex1 ex2) = tryOptimize (Add (evaluatePartially ex1)
    (evaluatePartially ex2))
evaluatePartially (Multiply ex1 ex2) = tryOptimize (Multiply (evaluatePartially ex1)
    (evaluatePartially ex2))
evaluatePartially ex = ex

-- Example Provided
exampleExpression = Add
                ( Add
                   ( Constant 20)
                   ( Constant 17))
                ( Add
                   ( Variable X )
                   ( Multiply
                      ( Add
                         ( Constant 14)
                         ( Constant 7))
                      ( Constant 2)))
```

HA 4