Programming Paradigms
Week 3

Jacek Wasilewski

Exercises

1. Define three expressions using `def`, `val` and `var`. Each expression should have a value of a different type, e.g. `Int`, `Double`, `String`. Do not specify type this time. Check if types have been inferred as expected.

2. Create another 3 expressions and explicitly specify type of each. Check if you can assign values that does not match the types.

3. Try to modify each of the expressions defined in 1. Is it possible?

4. Define a function taking two parameters `x` and `y` of type `Int`. Function should return the average of these two parameters. `Double` should be a returning type. Check correctness on few examples.

5. Copy the function from 4 and change the evaluation type to *call-by-name*. Is there any visible difference?

6. Write a function `func` that one parameter. Function should return number 42 (we do not want to use the parameter).

7. Define the following expression:

   ```
   def loop: Int = loop
   ```

   Then call function `func`:

   ```
   func(loop)
   ```

   What happened? (if it hanged, you must stop the worksheet manually)

8. Modify the function defined in 6 to use call-by-name parameter evaluation. Call the function with `loop` as parameter again. Do you see the same behaviour?

9. Write a function `term` that takes one natural number. If that number is even, function should return quotient of division by 2. If odd, multiplication by 3, plus 1.

10. Write a recursive function that:

    (a) takes a natural number `n`

    (b) if `n` is 1, returns `n`

    (c) if `n` is not 1, calls itself; as the input, output of function `term` on `n` should be used.

11. Modify the function from 10 to print the current value of `n` before the recursive call. You can print out using `println` function.

12. Function `term` is only used by the function from 10 and we do not want to expose it. Move it appropriately.

13. Write a function that calculates the product of all natural number from 1 to `x`. `x` is the parameter of that function.