Programming Paradigms
Assignment 1

Jacek Wasilewski

1 Description

Solve problems defined below. You should submit one Scala Worksheet file (`.sc` file extension) containing all your work. Separate and comment each problem appropriately. Include only code that compiles - any compilation error will result in 20 points penalty. Totally you can get 100 points, problems are not equally weighted. Submit before the deadline stated on Moodle.

2 Problems

2.1 Problem 1 (10 points)

Write a function for the second degree polynomial. Second degree polynomial is of form:

$$y = ax^2 + bx + c$$

Your function should be named `second` and it should take 4 parameters: a, b, c and x. Parameters should accept real numbers and a real number should be returned. Write an anonymous version of this function.

2.2 Problem 2 (15 points)

Given are the following functions:

```
def g(x: Double) = x + 1
def h(x: => Int) = g(x) * x - 1
def f(x: => Double, y: => Int) = g(x) + h(y)
```

Show the step-by-step parameters evaluation and functions re-writes for the following function call:

```
f(2.0 + 3.0, 2 * 2)
```

Put the evaluation in the worksheet as a block of comments.

2.3 Problem 3 (15 points)

Write a recursive function that calculates n-th Fibonacci number. Function should not be tail-recursive. Function should be named `fibonacci`, it should take one parameter of type `Int`.

The n-th Fibonacci number is defined as follows:

$$F_n = F_{n-1} + F_{n-2}$$

and

$$F_1 = 1, F_2 = 1$$

First few Fibonacci numbers are:

```
1, 1, 2, 3, 5, 8, 13, 21, ...
```

2.4 Problem 4 (25 points)

Given is the sequence defined as:

$$\begin{aligned} a_1 &= 3 \\ a_2 &= 5 \\ a_n &= (-1)^n \cdot 5a_{n-1} + (-1)^{n-1} \cdot 3a_{n-2} \end{aligned}$$

a) (20 points) Write a function that returns the n-th element of the sequence. Function should be named `series`, should take only one parameter and should be implemented recursively, in a tail-recursive manner.

First few elements of the sequence are:

```
3 5 -16 -105 ...
```

b) (5 points) Prove that function is tail-recursive.

2.5 Problem 5 (35 points)

a) (20 points) Implement function `higher` that takes 4 parameters: f, g, x and y.

Parameters f and g are functions. Function f takes 2 integers and returns a boolean. Function g takes 1 integer and returns a boolean. Parameters x and y are integers.

Function `higher` should check the following expression and return its result:

$$f(x, y) \wedge (g(x) \vee g(y))$$

b) (5 points) Show the usage of the `higher` function using anonymous functions equivalent to the following named functions:

```
def greaterThan(a: Int, b: Int) = a > b
def greaterThan0(x: Int) = x > 0
```

c) (10 points) Create a curried version of the `higher` function. Name it `curriedHigher`. Separate types that are functions from the integers. Provide examples of execution.