Exercise 5 – Description

Problem 1: Extend the class IntList with the following the functions. Implement the function in the class IntList as well as in the concrete classes Cons and Empty.

a) A function map(func), that applies a function func to all elements of the list and returns a list containing the new values.
b) A function filter(func) that filters the list regarding a given predicate and returns a new list containing all elements that fulfill the predicate.
c) A function foldLeft that aggregates a list from left to right starting with a base element. Use the following order: op(... (op(op(base, a_0), a_1) ...), a_n)
d) A function foldRight that aggregates a list from right to left starting with the base that need to be combined with the last element. Use the following order: op(a_0 op(a_1, op(..., op(a_n, base)...)))
e) A function reduceLeft that aggregates a list from left to right starting with the first element that need to be combined with the second element. Use the following order: op(... (op(op(a_0,a_1), a_2) ...), a_n). The function reduce has no base element.
f) A function reduceRight that aggregates a list from right to left starting with the last element that need to be combined with his predecessor. Use the following order: op(a_0 op(a_1, op(..., op(a_n, an-1)…))).

Use only means of functional programming – only immutable variables and recursions.

Problem 2: Complete functions in the class Application. Use only Higher Order Functions for your solution.