## Scala in Practice

## lab 03

#### Acceptance criteria:

Create Scala program with:

• **object** Utils implementing methods:

def isSorted(as: List[Int], ordering: (Int, Int) => Boolean) = ???
//checks if as is sorted according to ordering

def isAscSorted(as: List[Int]) = ??? //checks if as is sorted in
ascending order

def isDescSorted(as: List[Int]) = ??? //checks if as is sorted in
descending order

**def** foldLeft[A, B](l: List[A], z: B)(f: (B, A) => B): B = ??? //applies binary operator to a start value and all elements of l, going left to right. Dont use build-in List.foldLeft<sup>1</sup>

def sum(l: List[Int]) = ??? //sum elements of l with usage of
foldLeft function

def length[A](l: List[A]) = ??? //length of l with usage of
foldLeft function

**def** compose[A, B, C](..., ...) = ??? //compose two unary functions:compose(f,g)(x) = f(g(x)). Dont use Funtion1.compose<sup>2</sup>

**def** repeated[A, B](..., n: Int) = ??? //takes unary function f with integer n & returns the n-th repeated application of the function. For example: repeated(f, 3) = f(f(f(3)))

**def** curry[A, B, C](...) = ??? //converts a binary function f of two arguments into a function of one argument that partially applies f. For example, when def add(a: Int, b: Int) = a + b, then curry(add)(1)(1) == add(1, 1)

**def** uncurry[A, B, C](...) = ??? //reverse of curry function. For example, uncurry(f)(1, 1)(1) == f(1)(1)

<sup>1</sup> https://www.scala-lang.org/api/2.13.0/scala/collection/immutable/List.html

<sup>2 &</sup>lt;a href="https://www.scala-lang.org/api/2.13.0/scala/Function1.html">https://www.scala-lang.org/api/2.13.0/scala/Function1.html</a>

# **Scala in Practice**

### lab 03

• Create function to *standardize* handling of unsafe parts of code - the ones which could throw exception.

```
def unSafe[T](ex: Exception)(. . .) = ??? //catch any exception, log
the error & throw the ex exception instead

Now, in the code we could have:

unSafe(MyException("Could not run command")) {
    //block of code which could throw any exception
}

. . .
unSafe(MyDifferentException("Error while building map")) {
    //block of code which could throw any exception
}
```

• Create *application entry-point* object with some example tests for the above implementation

Michał Kowalczykiewicz