Exercise 3:
Liskov Substitution Principle

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Engineering Design & Construction
WS 2016/17 - Dr. Michael Eichberg, M.Sc. Matthias Eichholz

Although the other exercises are not graded, it is highly recommended to also do them on your own. Just looking at a solution is much easier in comparison to actually coming up with it. Support can be found in the forum:
`https://www.fachschaft.informatik.tu-darmstadt.de/forum/viewforum.php?f=234`

Introduction

You will implement a small class hierarchy for immutable tuples in Scala and Java. A tuple is a sequence of elements of a fixed length. The base trait/fully abstract class `Tuple` should take one type parameter `A` that denotes the type of elements the tuple can contain. Every tuple of length n should expose the following methods:

- A `length` method returning the number of elements n in the tuple.
- A `get` method, returning the element at a given index (starting at 0). It throws an exception if the index is < 0 or $\geq n$.
- A `contains` method that checks whether a given object is an element of the tuple.
- An `add` method that creates a new tuple of length $n + 1$ that contains the elements of the existing tuple with a given element appended.
- A `map` method that executes a function on each element of the tuple.

You should implement concrete subclasses `Tuple0`, `Singleton`, `Pair` and `TupleN` for tuples of length 0, 1, 2 or any number ≥ 0, respectively. Their constructors should take the corresponding number of elements. For `TupleN`, you are free to use varargs, an array or some collection.

Task 1 Scala

Implement the 5 tuple classes and 5 methods in Scala. First, **use variance annotations** + or - for every type parameter for which they are possible. Second, the **types of the methods should be as precise as possible**. Third, please implement the `get` method as `apply` method. In particular, the `contains` method should have a parameter with a type more precise than `Any` or `Object`.
Can you define `Tuple0` as a Scala **`object`** instead of a Scala **`class`**? If you can, simply replace the class definition with a corresponding object definition. If you cannot, leave a comment why.
According to the Liskov Substitution Principle, could any of the concrete tuple classes inherit from each other? If yes, leave a short comment how, and why another way is not possible. If you don't think it is possible, also leave a comment why. (Please don't try to implement it, though).

Task 2 Java

Implement the 5 tuple classes and 5 methods in Java. Since Java has use site variance, you cannot use variance annotations. However, again, the **types of the methods should be as precise as possible**. They will be different (potentially less precise) from the Scala solution, though, but the `add` method should not contain imprecise types such as `Object`. Hint: you are free to make the `add` method static in order to achieve this.