

## Triangle Report

Group FT\_I

Dominik Safaric, Nian Liu, Guy Rombaut, Bas Meesters

The assignment itself did not take a lot of time. In a few minutes each of us was able to implement the different rules for triangles in Haskell. After some improvements when comparing the result we could work on the testing part. This was actually a lot harder since we were a bit stuck about how to properly test it. Even if every input of three integers will give a shape, we don't know anything about the correctness of this output. After some council from Vadim we chose to mainly use list comprehensions for testing. We generate a lot of triangles on which we can reason about. For example, when two of the three sides are set and the third side is increased every iteration there should be exactly two Isosceles among them when both side values are passed since both sides have the same value as the third one at that moment. The same can be similarly done with NoTriangle, Equilateral and Rectangular. We counted each of the different shapes and were confirmed that the amounts of specific triangles were the same as expected. Thus concluding that the function gives the expected results for a lot of different values. Of course, we only tested a few cases and cannot for certain say that the function is entirely correct.