Unit testing of the polygon function

The objective of the polygon function is to check how many entries are there in the list and then to call the appropriate function if necessary.

The polygon(nodes) function takes as input a list of integers. Thus out of {int,float,string,list} the first three are invalid types. We create one test for each of these and assert that the polygon function returns “Input is not a list”.

We now turn to the valid inputs. The input domain can be divided into four equivalence classes (EC):

1. length of list is at most 2,
2. length of list is 3,
3. length of list is 4, and
4. length of list is strictly more than 4.

The coverage criteria we will use are:

For each EC, we need to test at least one input.

This leads to the following test requirements:

R1. If input is in EC1, polygon should return “Not a polygon”

R2. If input is in EC2, polygon should call the triang function

R3. If input is in EC3, polygon should call the quadrilateral function

R4. If input is in EC4, polygon should return “Larger Polygon”

These lead to the corresponding test cases:

TC1. Input=[1]

TC2. Input=[1,2,3]

TC3. Input=[1,2,3,4]

TC4. Input=[1,2,3,4,5]

In order to be able to execute TC2 and TC3, the fixtures triangle\_stub1 and quadrilateral\_stub1 were written.