Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

Exercise Sheet 2

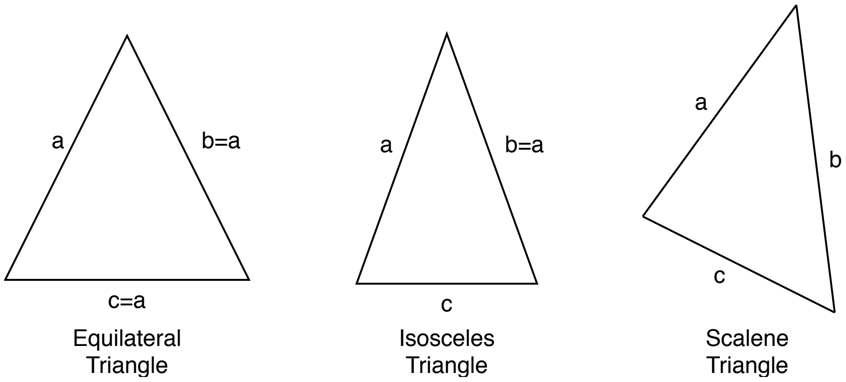
Software Architecture for Distributed Embedded Systems, WS 2021/22

Prof. Steinhorst, M. Sc. Emanuel Regnath, M. Sc. Jens Ernstberger

Exercise 2.1: Just a Triangle

After desperately trying to find your first customer as a small team of developers, you stumble upon a customer with an ambitious software project in mind. The customer gives you this requirement:

> I want a software that can detect different types of triangles. Whenever I provide it three numbers, let's say a,b and c, it should output equilateral, isosceles or scalene triangle. Like in the following figure!



Software development requires planning and execution of this planning in order to build large applications, generally referred as software development cycles. We will do a very quick software development using the V cycle (modified waterfall cycle) (even if seems like an overkill *in the beginning*), starting from these customer requirements. *There is one rule: You are not allowed to write code unless it is explicitly allowed!*

1. Starting from the customer requirement, what are the next steps before the software is delivered to the customer?
2. What are the analyzed software requirements? Generally the customer requirements can be vague where you need to find the actual requirements to start implementing. These analyzed requirements can be considered as the design documents.
3. Write a Python script or pseudo code, or even logic, according to these requirements. You can either assume that the numbers are passed as arguments or the script prompts the user to input numbers. *You are allowed to write code here.*
4. How can you rewrite the code if you had the requirement that the code has to be object oriented?
5. Write the tests that will test the code based on your analyzed requirements. *Do not* look at your code as these should be based on the analyzed requirements.

6. Once your tests are passed, you ship the software the customer. What happens when the customer puts the following values into the code (will be revealed during the session)? Do you have any unexpected behavior? If yes, what needs to change **first**?
7. Identify the source of each unexpected behavior? (E.g. customer requirements, analyzed requirements, user, code)
8. What would be the new analyzed requirements?
9. How would this development change in an agile approach?