Author: Andreas Traut
Date: 22.02.2021
Download as PDF

Algorithms, Data Structures and Coding

0. Introduction

a) Aim of this repository

The aim of this repository is to share my coding skills, knowledge in data-structures (e.g. classes), abilities in algorithmic thinking (e.g. recursion) and tool-building skills (e.g. Excel/VBA tools). At the same time I will give you lots of hints and solution templates, which will help you to enhance your skills in these topics as well.

b) Motivation for this repository

I am programming in different languages and environments for nearly my whole life:

- Starting in the 1980s / 1990s with GW-Basic and the integrated development environment (IDE) Turbo Pascal.
- Then in the 2000s / 2010s I started with C++, where I understood the **object oriented way of thinking**. I learnt a lot in C++ at my final years at university as well as during the first years at my first employer. Today my 8 cm thick *"C++ programming bible"* serves as an elevation for my second monitor, which I had to set up due to the Corona-related home office.
- I used SQL a lot and also got quiet skilled in finding solutions with Visual Basic (VBA). VBA (applied in Excel or Access) is fun for me and served me a lot during my whole professional and private.
- In 2019 I learnt the advantages of the Jupyter-Notebooks: beautiful, intuitive, easy to use and build. But there is something, I don't like in Jupyter-Notebooks, which I will exlain below.
- And today I am a big fan of Python: it's so much more fun to use Python instead of C++: I enjoyed not having these opening brackets `{` and closing brackets `}` and `;` a the end of a line! Such a relieve for my eyes.

I am glad, that lots is similar in all these decades: **the way of thinking as a programmer**. My motivation is to give you some basic hints, advises and guidance to improve your coding skills.

c) Structure of this repository

(i) First part: How to improve your coding skills: Certificates and Challenges

In the *first part* I will explain, how certificates and coding challenges can be useful for you to improve your coding skills.

(ii) Second part: Examples

In the *second part* I will work on some interesting examples.

I. How to improve your coding skills: Certificates and Challenges

1. Earn a certificate

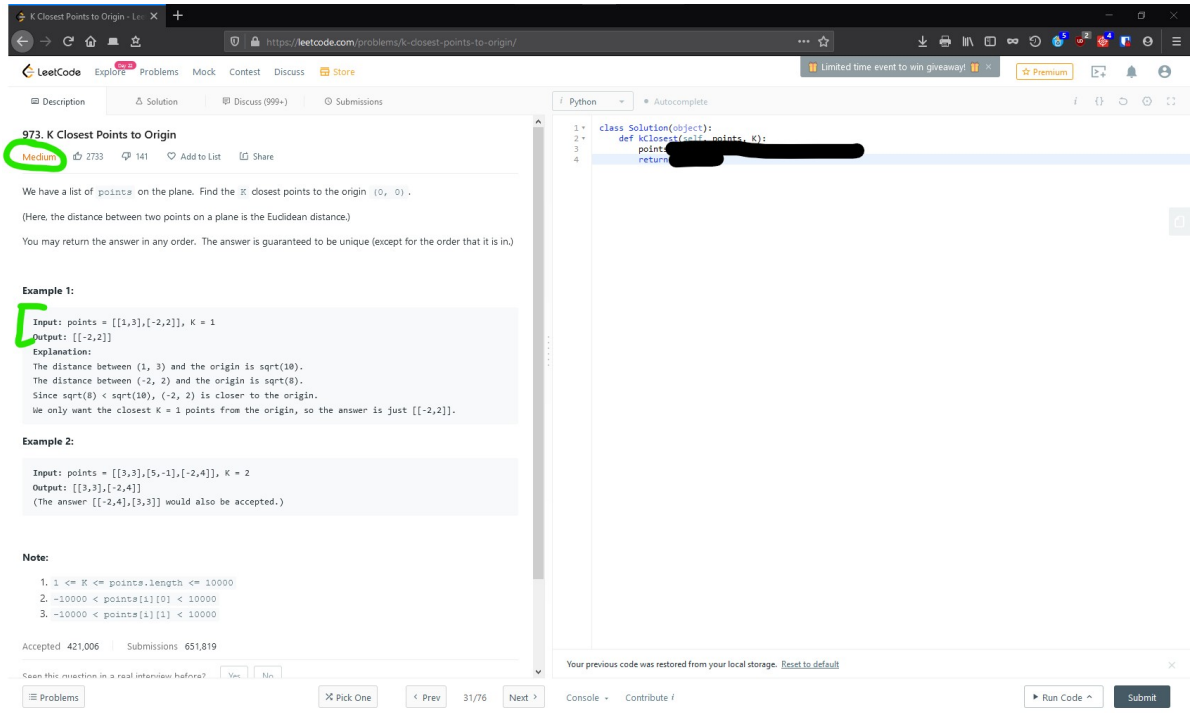
A good way for improving your coding skills are by going through some online courses and trying to earn a certificate. There are a lot of other resources: maybe start getting an overview on Coursera. These courses are nice because the teachers are usually highly skilled (from universities) and the technical infrastructure for the courses is rather advanced: there are videos with subtitles and transcript and you can easily navigate through these videos by reading across these transcripts and jumping to the positions in the video, which you want to listen to. You can monitor your learning curve and weekly progress. But the Coursera certificates usually cost some money.

If you want to find something cheaper, then I can recommend the "Data Structures and Algorithms in Python" from Jovian. When I worked for it in 02/2021 is was for free. It uses Jupyter-Notebooks and is definitively a lot of fun! You will learn in video tutorials and practice with well documented Jupyter-Notebooks how to work with python classes, binary trees, sorting algorithms and understand how to solve coding problems systematically.

There are various other resources for earning a certificate and listing up, what I found is not very helpful at the end for you: try to find **the certificate which YOU want to earn!** I promise: working for it is a lot of fun.

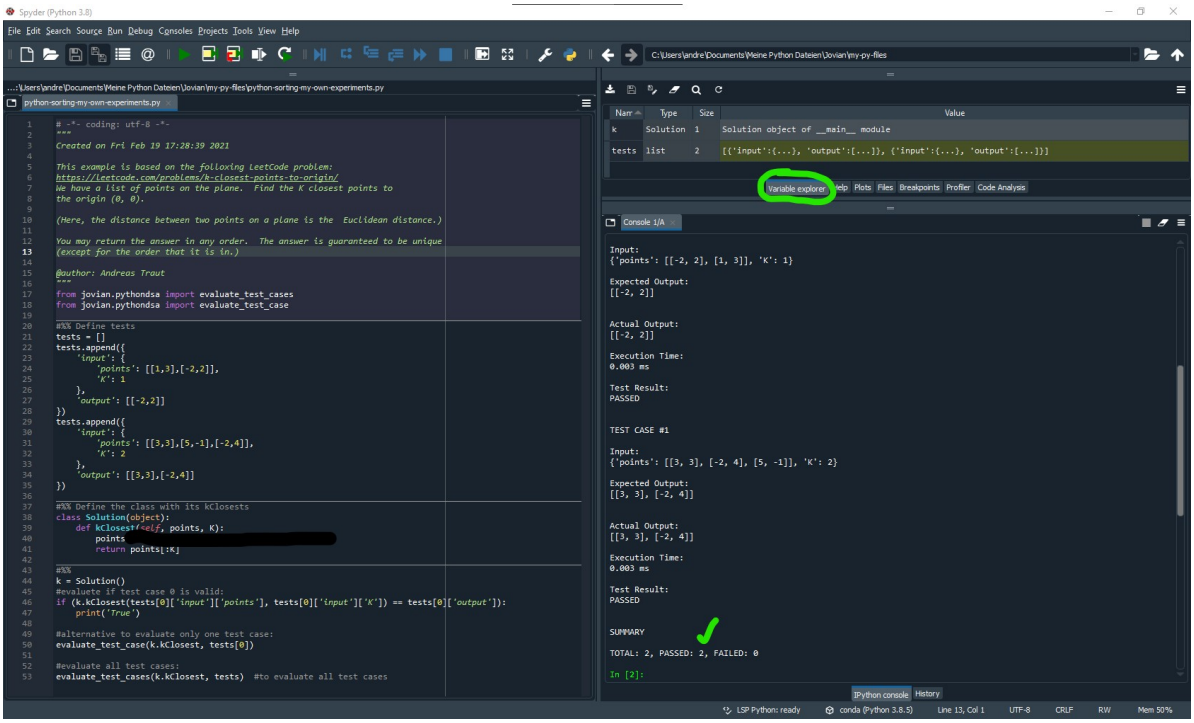
2. Get into coding challenges

Another advice I can give you is to get into coding challenges. When you accept a coding challenge, then a problem will be shown and would have to solve it in your preferred programming language (python, java, C++,...). I tried LeetCode and you will find a lot of other websites, which provide similar concepts. On the left is the problem, on the right some place to program a solution:

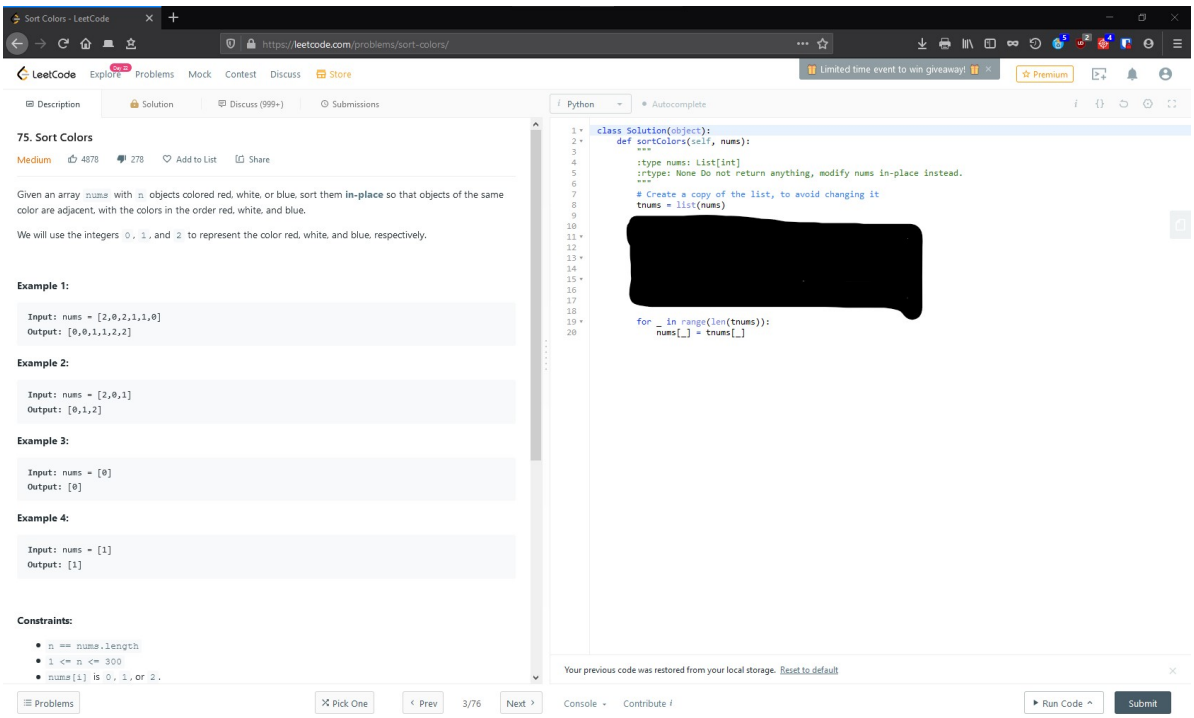


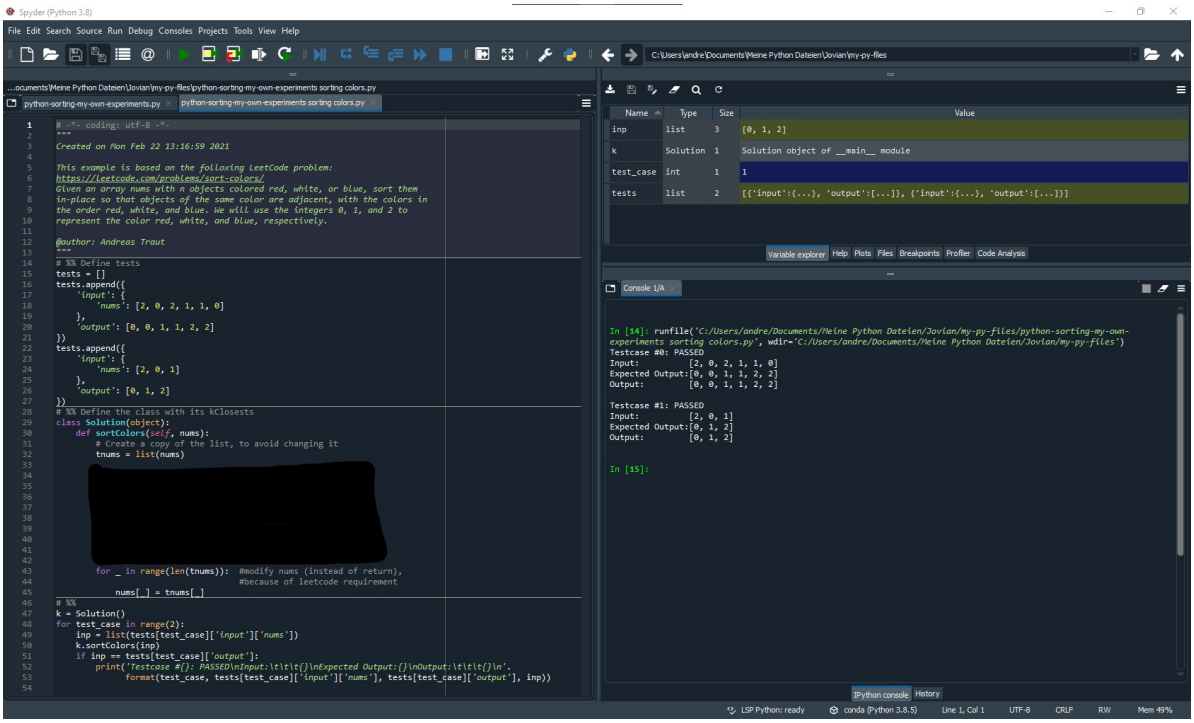
As I am not allowed to publish solutions for these LeetCode problems I had to black out my solutions. Some of these problems were quiet interesting for me so I wanted to have them in my integrated development environment (IDE) Spyder-IDE in order to debug through the code and extend the examples a bit. I recommend to use an integrated development environment (IDE) as often as you can, instead of always going through Jupyter Notebooks. In my opinion Jupyter Notebooks are **not** always the best environment for learning to code! I agree, that Jupyter Notebooks are nice for doing documentation of python code. It really looks beautiful. But I prefer debugging in an IDE instead of a Jupyter Notebook: having the possibility to set a breakpoint can be a pleasure for my nerves, specially if you have longer programs. Some of my longer Jupyter Notebooks feel from the hundreds line of code onwards more like pain than like anything helpful. And I also prefer having a "help window" or a "variable explorer", which is smoothly integrated into the IDE user interface. And there are a lot more advantages why getting familiar with an IDE is a big advantage compared to the very popular Jupyter Notebooks! I am very surprised, that everyone is talking about Jupyter Notebooks but IDEs are only mentioned very seldom. But maybe my preferences are also a bit different, because I grew up in a MS-DOS environment. :-)

Here is how the problem from above looks like in the Spyder-IDE:



Another example from LeetCode: the [Sort-Colors](https://leetcode.com/problems/sort-colors/) problem:





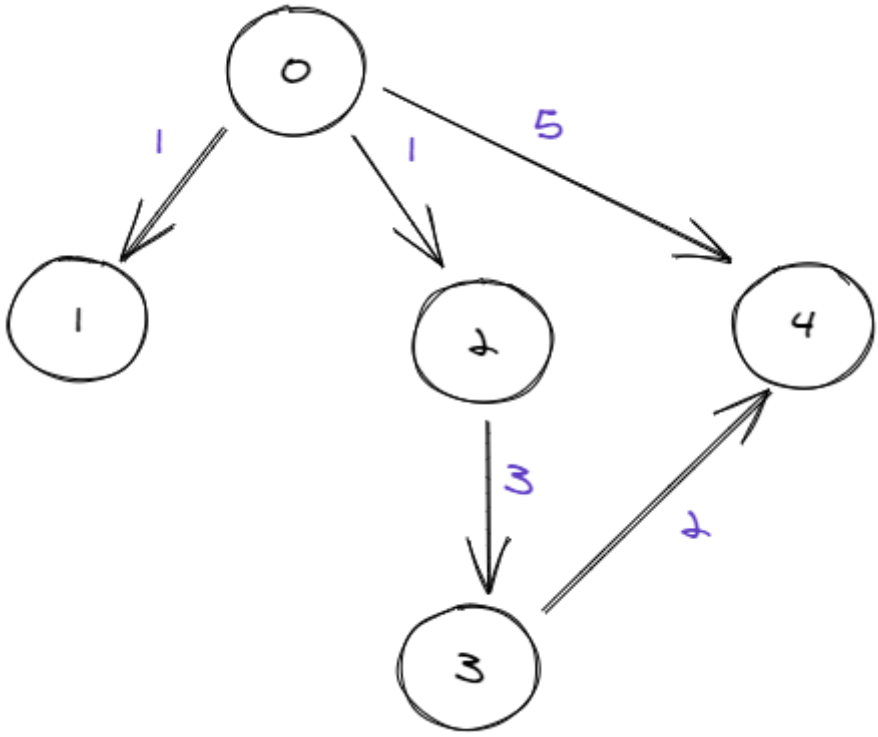
II. Examples

In the *second part* I will work on some interesting examples, which will be available as `.py` Python-Files, Jupyter-Notebooks or will be Tools like Excel/VBA or Access.

1. Python-Examples

I solved several problems, which require algorithmic thinking (e.g. recursion) or knowledge about data-structures (e.g. casses) and I shared my solutions as `.py` Pyhton-Files here. These problems are for example:

- Find the K nearest points to the origin, given some points in the plane.
- Sort the list of color-codes.
- Calculate the delay time for a signal, which is send over a network graph.



Each time I will build some test-cases before going into the solution and also use the Jovian "evaluate_test_cases" module to efficiently perform the tests.

You can download this examples from my repository:

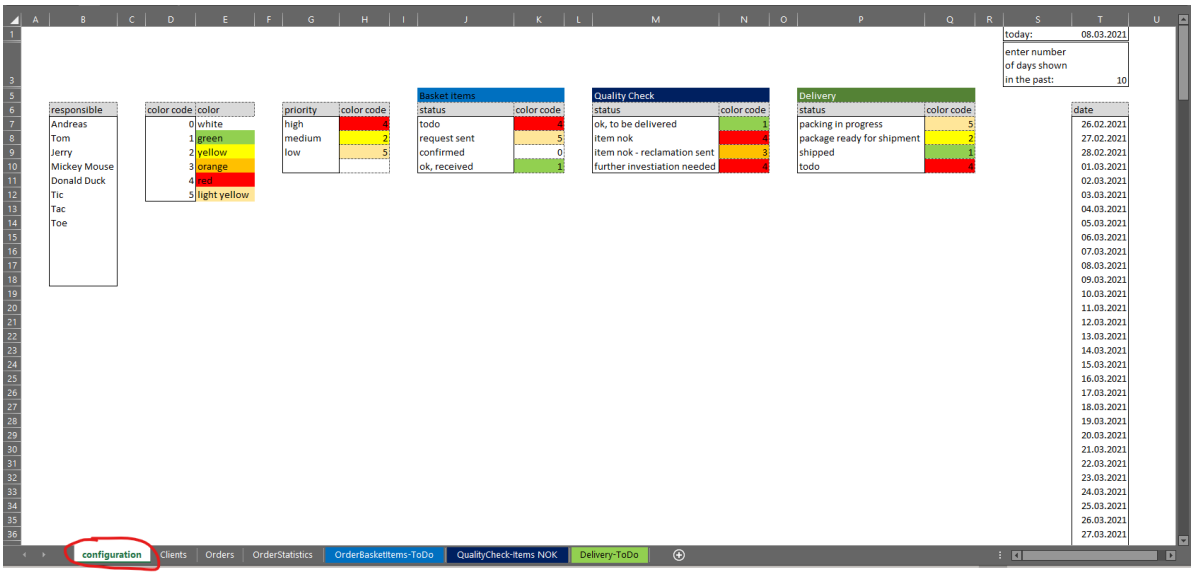
https://github.com/AndreasTraut/Algorithms-Data-Structures-and-Coding/tree/main/Python_Examples

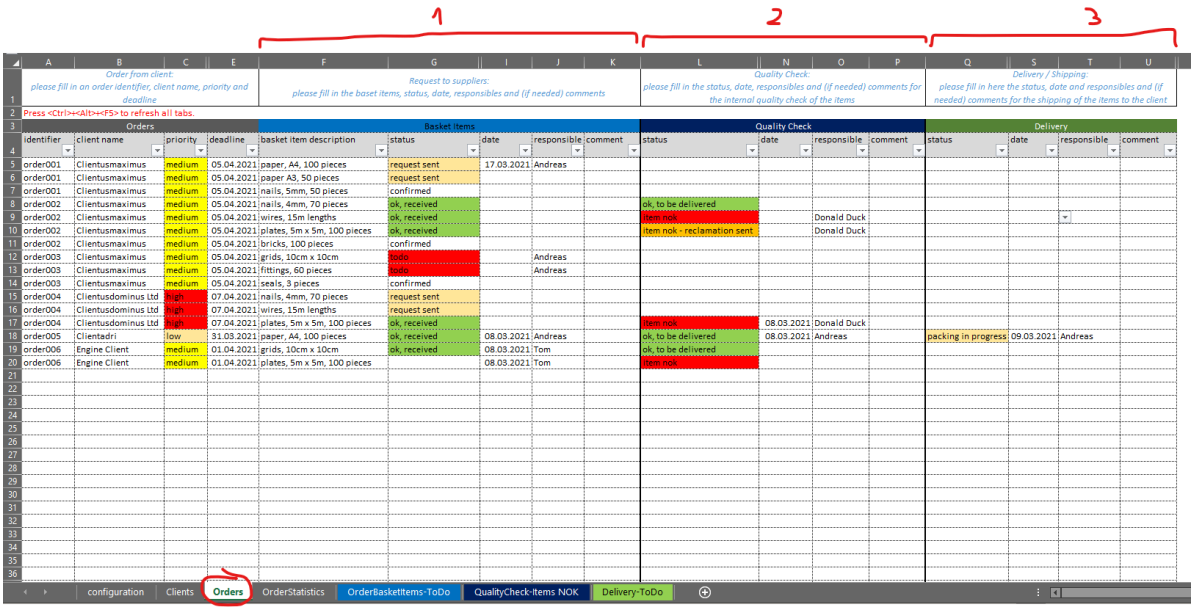
2. Excel Example

During my career I implemented a lot of Excel/VBA solutions: one was a Excel/VBA project management tool, which organized and structured a complex project flow of a team of 7 people. My Excel/VBA solution is used on a daily basis and is running for already 2 years now.

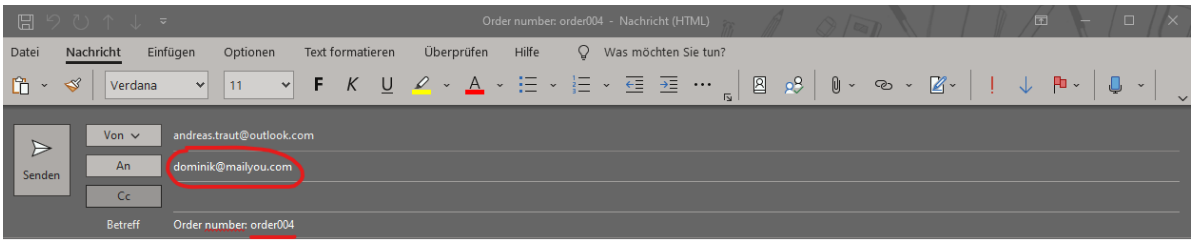
I won't be able to mention all the other Excel/VBA which I built or worked on and I also won't be able to share my Excel/VBA tools here, which I implemented at different companies due to copy-right restrictions. But I will provide an example of an Excel/VBA solution, which solves the following order tracking problem: assume, that you are responsible for different clients, which order different items from you. Each time they do, you would have to send requests to your suppliers (see "1" in the screenshot below) . After having received the items from your supplier you will do an internal quality check (see "2" in the screenshot below) and then send the items to your client (see "3" in the scrnshot below). You and your team colleagues may want to track all the different items and also the cases, when something went wrong (item not yet received, item did not pass the quality check,...).

The first step is to define the three steps ("1. Basket Items", "2. Quality Check", "3. Delivery") and assure in the tab "configuration" that the predefined dropdown cells and color codes, are always **clear**. Like this your will get **consistency in your processes and data**. Changing the color codes or status description here will automatically update the whole Excel/VBA solution and therefore you will always have consistency.





Additionally you may want to inform your client about the intermediate status of their orders by automatically generated Outlook-Emails. I implemented this in Excel/VBA and pressing one button will create an Outlook-Email, where email-address, subject and email-text is filled automatically by my VBA code as follows:



Dear Dominik,

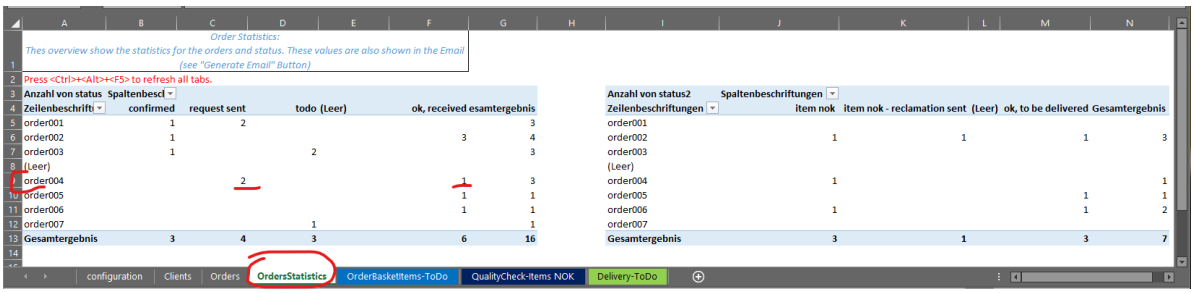
I would like to inform you about the current status of the order number order004:

1 out of 3 items have already been delivered by our suppliers and are ready for our internal quality check
0 out of 3 items have successfully passed our internal quality check and are ready for being shipped.

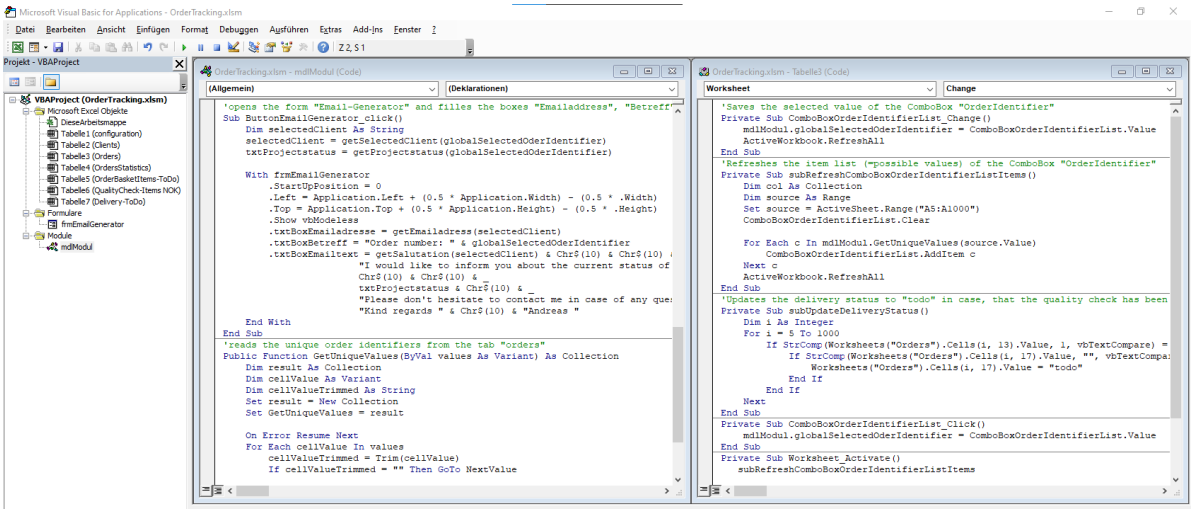
Please don't hesitate to contact me in case of any questions.

Kind regards
Andreas

Furthermore some statistics should help you to see, where you have issues in your order process (like failed quality checks,...):



Let's have a short look into the VBA code:



You can download this example from my repository:

https://github.com/AndreasTraut/Algorithms-Data-Structures-and-Coding/tree/main/Excel_Example

3. Access-Example

During my career I also worked with Access solutions. One was for a team of 20 people who were working simultaneously with their Access-frontends on one Access-backend. The aim was to assure a structured data-entry of the whole team into the Access backend database by using Access-Forms. At the end of the project the filled database tables had been joined with SQL queries to further databases in order to aggregate a very specific result table.

I also worked on other Access solutions and will provide an example of an Access solution in short time here.

MIT License

```
MIT License
https://opensource.org/licenses/mit-license.php
```