# Task Clip:

This Task is meant to be solved using python, contributing a python script.

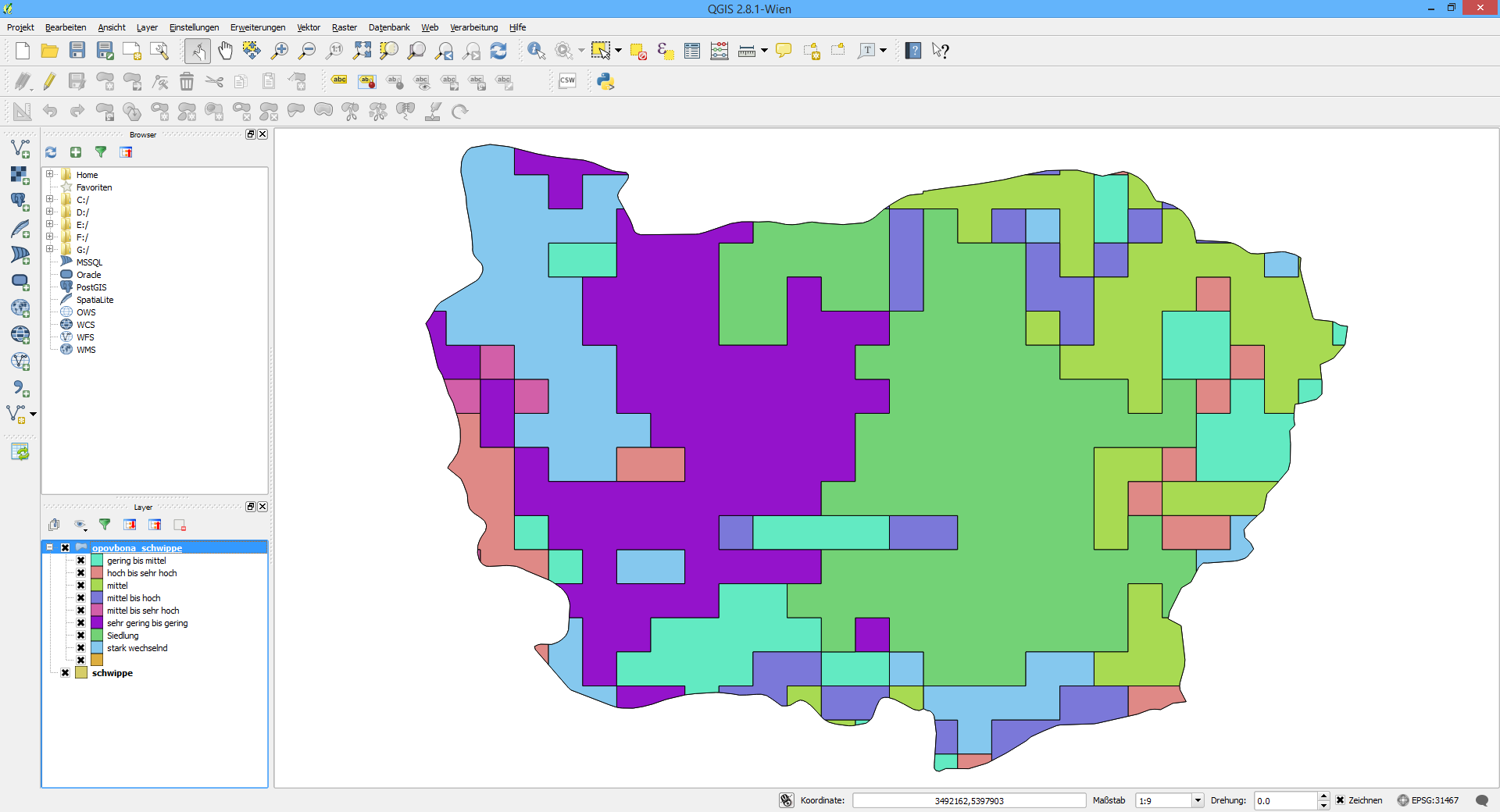
The WaBoA, the Wasser-Boden Atlas Baden-Württemberg, is a collection of measured and modelled geodata for Baden-Württemberg. It contains general catchment parameters as geology or natural vegetation.

In this task, you are only interested in the data relevant for your specific catchment, which needs to be extracted. In order to do so, one will have to run the same procedure repeatedly for any parameter used. In order to fulfil this task, write a python script supporting that task.

Three subfolders containing all necessary data are contained in the task folder. Each one contains a shapefile for the catchments themselves, geology and natural vegetation, respectively. Your catchment is a sub-catchment of the Neckar River with an extent of 86 km². The catchment is called ‚Schwippe‘.

You will get an A in case your script can handle a varying amount of input shapefiles (Imagine you have 100 shapefiles). An you will get an A+ in case the subcatchment to be choosen can be changed with only one line and the script can also apply other functions like creating a new field containing the distance to the catchment centroid.

If you open the result in QGIS and choose any random category-styling, it will look like this:



The A++ is for those, who contribute a script that can clip a varying amount of subcatchments from a varying amount of shapefiles and adding a field for „distance to University Freiburg“ and „Area“ for each feature.

**NOTE:** The data from WaBoA is **NOT** free to use. **By this remark, it is officially stated that there is no permission for data usage outside the course**.

**NOTE:** Any uncommented python code will be completely ignored. So please use comments extensively!

**NOTE:** You are not allowed to use command line tools like ogr2ogr.

Some tips:

Make it clear to yourself what you want to do, maybe give it a first try by using QGIS.

All WaBoA data uses Gauss-Krueger Zone 3 projection. In case this information is missing for some shapefiles, find out the EPSG number and handle this error.

Create a useful folder hierachy. Give useful names to all created files.

Some usefull packages:

* glob: supports unix styled file path handling like \*.shp
* re: regular expressions, eg. Make *somethnig*.shp to *something*\_output.shp
* os: operating system functions like change directory, check if a file exists or make new folders

Read about ogr.Layer objects filtering options!

To find some help have a look at:

https://pcjericks.github.io/py-gdalogr-cookbook/vector\_layers.html#filter-by-attribute

http://pcjericks.github.io/py-gdalogr-cookbook/layers.html l#filter-and-select-input-shapefile-to-new-output-shapefile-like-ogr2ogr-cli