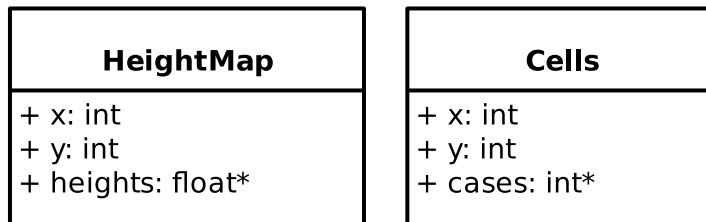**Group: Trygve and Esther**

Functionality

Our program usees the marching squares algorithm to create a vector contour map from a raster heightmap. The cli interface is `gdal_contour [OPTIONS] <src_filename> <dst_filename>` with these options:

```
-i <elevation intervall>     Interval between contours
-f <format>                  Fileformat to output
```

data structure and input/output

HeightMap
+ x: int
+ y: int
+ heights: float*

Cells
+ x: int
+ y: int
+ cases: int*

Both will need to implement the rule of three because they store raw pointers to arrays

\# Responsibilities: Esther will create the algorithm itself with multitreading. This will essentially be a function that takes a grid of pixels as input and returns a similar grid of cells. Trygve will take care of reading in the tiff file into our own datastructure and creating a vector image from the output of the algorithm.

How do you plan to make it easily verifiable that your objectives are reached?

We can compare against the `gdal_contour` cli program which is a implementation widely used in other software. We can compare speed, memory usage and the result itself. Each step in our program also produces a output which we can be worked on and evaluated independently.