## OOADI-møde

Looks like a good design

Be careful about how to create response and request packets. Think about the design of these classes. Make everything we send back and forth serializable.

Clients: Requests for add/remove plant and change conditions must be serializable

Server: Logs (easy as it is text) and plants must be serializable

Create an overview of how the data flows in the program before starting programming. Think about some use cases and create the overview based on these. Example raise temperature: the change comes from the class X and Y, and it goes through the program like Y.

Once we know the main cases, does the architecture then allow you to perform these use cases. Related to the time at which we discover a bug. It is easier to fix it, the easier we catch it.

The solutions to the exercises will be uploaded over the next couple of meetings.