

## Seminar 1 in TDDB84 - Use cases for design patterns

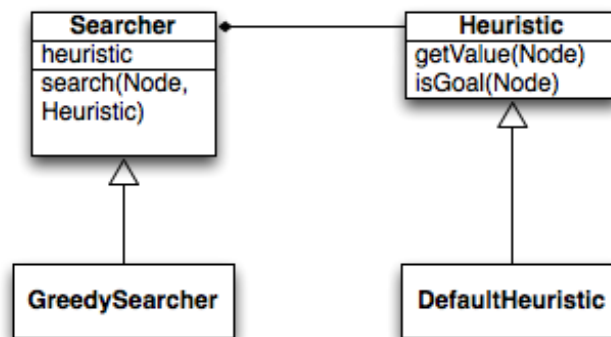
Ola Leifler - 2017-05-12

Choose one design pattern for each case, and justify by showing an example UML diagram and a justification. The selected design pattern shall describe the main functionality requested, not a partial implementation detail.

A use case may be described as follows:

*Describe how to customize search methods to find the safest or fastest way through a city. Assume that there is a class Searcher that performs the general search method. Basically, the Searcher will accept a data structure that represents the road network in the city in the form of a graph, and allow you to select how to evaluate the proximity of each node to a goal.*

This description could result in a UML diagram consisting of an interface Searcher with at least one method search, accepting a node and an object that can be used to assess the proximity of a given node to a goal. Such an object could be called a *search heuristic*, and would need at least a method getValue to assess the value of a node for comparison against other nodes when selecting which one to take next in the search for a goal node. The basic structure might look something like the picture below, where a Searcher uses a Heuristic object to check the values of each node, and then a GreedySearcher implements the search method in its own way (greedily searching, one would guess). No more details than this would be required. The design pattern below is called *Strategy*.



When you propose your design, you shall provide a short textual description of your proposed design along with a UML diagram as the one above. The UML diagram shall contain the main components of your design, including

- The names and inheritance relations of the types involved
- attributes and composition relationships of the types involved,
- method signatures for publicly available methods in relevant types

We will discuss the selected solutions based on their appropriateness for the task, and the ability to modify solutions as necessary. The most common qualities that we wish to assess are the flexibility of the solutions, as in the extent to which one is able to change the solution to accommodate new needs, and the extent to which each component has a single, clear and logical responsibility.