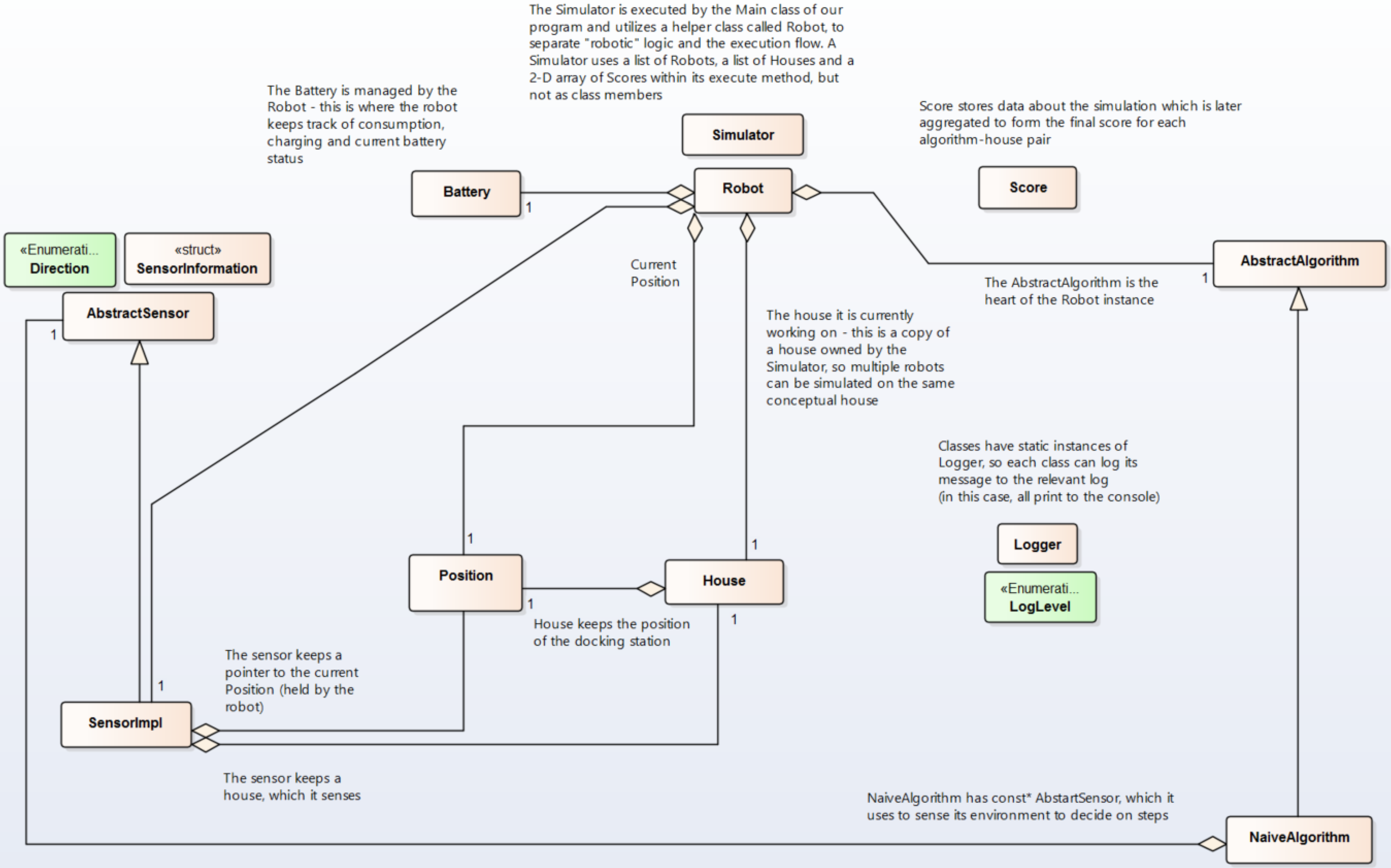

The Simulator is executed by the Main class of our program and utilizes a helper class called Robot, to separate "robotic" logic and the execution flow. A Simulator uses a list of Robots, a list of Houses and a 2-D array of Scores within its execute method, but not as class members

The Battery is managed by the Robot - this is where the robot keeps track of consumption, charging and current battery status

Score stores data about the simulation which is later aggregated to form the final score for each algorithm-house pair

The AbstractAlgorithm is the heart of the Robot instance

The house it is currently working on - this is a copy of a house owned by the Simulator, so multiple robots can be simulated on the same conceptual house

Classes have static instances of Logger, so each class can log its message to the relevant log
(in this case, all print to the console)

House keeps the position of the docking station

The sensor keeps a pointer to the current Position (held by the robot)

The sensor keeps a house, which it senses

NaiveAlgorithm has const* AbstartSensor, which it uses to sense its environment to decide on steps