Java Coursework, Group 28

Description of Design

Our design plan was to layout a skeleton code of each class required in the project, and then follow the design layout we had set for ourselves. We followed the design layout and divided the work between us by coding full classes (rather than just the skeleton class) separately and combining them together. This was possible due to successfully being able to follow the layout we had designed and set ourselves previously.

**Robot**

The Robot class declares multiple variables in the set up – it’s coordinates on the grid as integers, it’s UID (in order to create a new robot) as a String, a String item, a Boolean variable to check whether it has the collected item, a Boolean value for if it has or has not crashed, an integer value for the robot’s current charge, and it’s allocated charging pod set as the object ChargingPod made from the ChargingPod class.

The Robot class has multiple accessor methods, returning desired values. For example, getCharge() will return the current charge value of the Robot object. These are useful as they take affect and are required for the Simulator class in order to run the working model.

The ‘move’ method in Robot checks that the Robot has moved to new X and Y coordinates, and as a result of moving, removes a value of ‘1’ from its charge by calling on the ‘powerMinus’ method. This is necessary for the Simulator and Pathfinding Classes, as it needs the information of the Robots updated charge to work out whether it can make the trip to the Packing Stations on it’s current charge level or if it needs to return to the charging pod.

The Robot also continuously updates the item it is ‘collecting’, making use of the pickItem and drop methods.

**Storage Shelf**

The Storage Shelf declares a String integer

**Grid**