

Es wird zuallererst eine ArrayList importiert. Die Variablen „xPos“ und „yPos“, welche jeweils für die horizontale (x-Position im Koordinatensystem) und vertikale (y-Position) Position stehen, werden initialisiert und auf „0“ gesetzt. Die Variable „groesse“ speichert die Größe des Rasters, also die Anzahl der Zeilen und Spalten im Raster, und wird auf die Größe 1 (im Code: „groesse1“) gesetzt. Auch das Array „orte“ wird eingeführt, dessen Einträge den Ortsnamen oder „0“ (also leer) sein können, welches im Konstruktor der Fall ist. Nebenbei wird die Variable „blickrichtung“ eingeführt, welche die aktuelle Blickrichtung des Roboters (Norden, Süden, Osten, Westen) speichert, und im Konstruktor auf „N“ für Norden gesetzt wird. Die Methode „OrtHinzufuegen“ ermöglicht das Hinzufügen eines Orts mit bestimmten Namen an einer bestimmten Position (xPos, yPos) im Raster. Die Methode „StartPosiSetzen“ setzt die Startposition des Roboters, indem ein Ort mit dem Namen „Start“ gesucht wird, um die Position zu aktualisieren. Die Methode „StartBlickrichtungSetzen“ überprüft welche der Umliegenden Orte in Richtung Norden, Osten, Süden oder Westen liegt und legt die Blickrichtung anhand dieser fest. In der Methode „Vergleich“ werden die x- und y-Koordinate eingegeben und anhand dieser überprüft, ob sie mit dem Namen übereinstimmen. Dabei können die Koordinaten nicht größer sein als das Koordinatensystem (auch Raster). Die Methode „AusgabeList“ kann einen Wert aus einer ArrayList „liste“ an einer bestimmten Position „Stelle“ abrufen und zurückgeben. Die Methode „Richtungen“ nimmt eine Liste von Orten „Weg“ entgegen die der Roboter besuchen soll. Dabei wird die Liste „Richtungen“ initialisiert und setzt die Startposition und Startblickrichtung basierend auf den ersten beiden <orten im Weg. Der „switch“ Block verwendet die aktuelle Blickrichtung und überprüft, ob der Roboter in diese Richtung gehen kann. Beim Fall „N“ werden die Positionen yPos-1 (m = geradeaus), yPos-2 (m), xPos+1 (r = rechts), xPos+2 (r), xPos-1 (l = links) und xPos-2 überprüft. Beim Fall „O“ werden die Positionen xPos+1 (m), xPos+2 (m), yPos+1 (r), yPos+2 (r), yPos-1 (l), yPos-2 (l) überprüft. Beim Fall „S“ werden die Positionen yPos+1 (m), yPos+2 (m), xPos-1 (r), xPos-2 (r), xPos+1 (l), xPos+2 (l) überprüft. Beim Fall „W“ werden die Positionen xPos-1 (m), xPos-2 (m), yPos-1 (r), yPos-2 (r), yPos+1 (l), yPos+2 (l) überprüft.