

Production Line System

In a automated production line, **SortAgent** was tasked with one of the most critical jobs — identifying and sorting products based on their color. With a keen eye for detail and a clear set of rules, SortAgent ensured that the right products went to the right destinations.

Every item that passed through the line was analyzed, its color sampled, and evaluated. SortAgent used its internal logic to check if the color leaned toward a strong red, a prominent green, or a distinctive blue. Based on these observations, the agent quickly determined the appropriate category: **Red**, **SpoiledRed**, **Light Green**, or **Middle Green**.

When a product exhibited characteristics that didn't align perfectly with a single color, SortAgent swiftly declared it as **SpoiledRed or DarkGreen**, preventing defective items from progressing further. On the other hand, when products shone with consistent hues, they were swiftly labeled and sent to their designated areas.

In short, the products labelled as SpoiledRed and Dark green are sent to the Push agent. by calling **toPush** plan while the others are sent to the buildSection, by calling **toBuild** plan.

Rule Description	Condition	Action
Red 1	Red(high), Green(low), Blue(medium)	.print(" <i>RED1 – HIGH–</i> ")
Red 2	Red(high), Green(medium), Blue(medium)	.print(" <i>RED2 – HIGH–</i> ")
Red 3	Red(high), Green(high), Blue(low)	.print(" <i>RED3 – HIGH–</i> ")
Red 7	Red(medium), Green(high), Blue(low)	.print(" <i>RED7 – MEDIUM–</i> ")
Red 4	Red(medium), Green(medium), Blue(low)	.print(" <i>RED4 – MEDIUM–</i> ")
Red 9	Red(medium), Green(veryhigh), Blue(low)	.print(" <i>RED9 – MEDIUM–</i> ")
Red 10	Red(high), Green(veryhigh), Blue(low)	.print(" <i>RED10 – HIGH–</i> ")
SpoiledRed 4	Red(medium), Green(medium), Blue(medium)	.print(" <i>SpoiledRed4 – MEDIUM–</i> ")
SpoiledRed 5	Red(medium), Green(high), Blue(medium)	.print(" <i>SpoiledRed5 – MEDIUM–</i> ")
SpoiledRed 2	Red(high), Green(high), Blue(medium)	.print(" <i>SpoiledRed2 – HIGH–</i> ")
SpoiledRed 7	Red(medium), Green(medium), Blue(high)	.print(" <i>SpoiledRed7 – MEDIUM–</i> ")
SpoiledRed 3	Red(high), Green(medium), Blue(low)	.print(" <i>SpoiledRed3 – HIGH–</i> ")
SpoiledRed 13	Red(medium), Green(veryhigh), Blue(medium)	.print(" <i>SpoiledRed13 – MEDIUM–</i> ")
SpoiledRed 14	Red(medium), Green(veryhigh), Blue(high)	.print(" <i>SpoiledRed14 – MEDIUM–</i> ")
SpoiledRed 15	Red(high), Green(veryhigh), Blue(high)	.print(" <i>SpoiledRed15 – MEDIUM–</i> ")
Light Green 1	Red(medium), Green(ultramedium) ↓ Blue(medium)	.print("##### LIGHT GREEN 1 \$\$\$\$")

Light Green 2	Red(medium), Green(ultralow), Blue(medium)	.print("##### LIGHT GREEN 2 \$\$\$\$")
Light Green 3	Red(medium), Green(ultrahigh), Blue(medium)	.print("##### LIGHT GREEN 3 \$\$\$\$")
Light Green 4	Red(medium), Green(ultramedium), Blue(high)	.print("##### LIGHT GREEN 4 \$\$\$\$")
Light Green 4.1	Red(medium), Green(ultrahigh), Blue(high)	.print("##### LIGHT GREEN 4.1 \$\$\$\$")
Light Green 5	Red(high), Green(ultramedium), Blue(medium)	.print("##### LIGHT GREEN 5 \$\$\$\$")
Light Green 6	Red(high), Green(ultramedium), Blue(high)	.print("##### LIGHT GREEN 6 \$\$\$\$")
Light Green 7	Red(high), Green(ultrahigh), Blue(medium)	.print("##### LIGHT GREEN 7 \$\$\$\$")
Light Green 8	Red(high), Green(ultrahigh), Blue(high)	.print("##### LIGHT GREEN 8 \$\$\$\$")
Middle Green 1	Red(low), Green(ultralow), Blue(medium)	.print("ZZZZZZZZZZ MIDDLE GREEN 1 \$\$\$\$")
Middle Green 2	Red(low), Green(ultramedium), Blue(low)	.print("ZZZZZZZZZZ MIDDLE GREEN 2 \$\$\$\$")
Middle Green 3	Red(low), Green(ultralow), Blue(low)	.print("ZZZZZZZZZZ MIDDLE GREEN 3 \$\$\$\$")
Middle Green 4	Red(low), Green(ultramedium), Blue(medium)	.print("ZZZZZZZZZZ MIDDLE GREEN 4 \$\$\$\$")
Middle Green 1	Red(low), Green(veryhigh), Blue(medium)	.print("ZZZZZZZZZZ MIDDLE GREEN 1 \$\$\$\$")
Middle Green 1	Red(low), Green(veryhigh), Blue(high)	.print("ZZZZZZZZZZ MIDDLE GREEN 1 \$\$\$\$")
Dark Green 2	Red(low), Green(high), Blue(low)	.print("KKKKKKKKK DARK GREEN 2 \$\$\$\$")
Dark Green 3	Red(low), Green(high), Blue(medium)	.print("KKKKKKKKK DARK GREEN 3 \$\$\$\$")
Dark Green 3	Red(low), Green(veryhigh), Blue(low)	.print("KKKKKKKKK DARK GREEN 3 \$\$\$\$")

Recall Infos:

There is a plan namely `arrangeResources`, which runs the following statements. In Jason a plan is written as **"`!planName: condition <- action1(55); action2.`"**, for example **`+!openDoor: neighbour(true) & doorbell(true) & happiness (high) <- action1(55); action2.`**

A Message pattern is `.send(agentName,tell, belief(anyvalue)).`

A Message pattern is `.send(agentName,achieve, belief(anyvalue)).`

