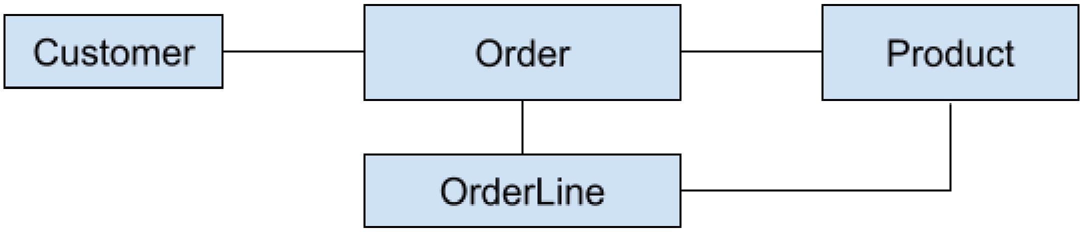
Part 1

You are designing and building the software controller for a food factory that has to control the ovens for multiple production lines.

A single controller is responsible for managing three ovens that cook different products that arrive in “assembly” lines, in a metal plate, let’s call it a “container”, having multiple items of a food product (e.g.: cookies, donuts, etc.). The different products that “arrive” at the oven, have different cooking times that have to be controlled, that is, each container has to be taken out when it is ready, this is done by an automated arm that takes the containers to cook from a line, and puts back elements when cooked. Each container also specifies the next assembly line that will be used to redirect it to an appropriate line (e.g.: a cookie product might need to be put in the line for the chocolate covering, while a donut might be put in a line to be injected with caramel). The controller has control over the assembly line to be able to stop it, the arm that takes products out and in, and the ovens. The oven has a maximum capacity to cook elements at a given time. The controller can signal the assembly line to stop, if the ovens are full.

Produce a conceptual class model that describes relationships between object types, no need to assign members at the moment, e.g:
Describe the relationship between the algorithm->model in the simple happy path scenario of two different food items (different cooking time) arriving in the assembly line of someone in L pressing the button up while the elevator is in 5.



Describe the relationship between the algorithm->model in the simple happy path scenario of someone in L pressing the button up while the elevator is in 5.

Part 2

After some testing, production control determines that the current line is taking too long to cook the different products, which makes the assembly line move too slow, and the cooking part is the slowest link. It resolves that the current disposition will be changed, three more ovens will be added, and four more arms.

At this point, our product specialist has added a new requirement: The arms will work for multiple ovens, and each oven will take products from any line. It will be possible to bring an oven down, without stopping the production.

Create a new version improving the previous model by adding its members (including data types) and honoring the requirement described previously. One example of such diagram can be found in figure 3.

You can use as data types any construction simple or not, present in any of the most common programming languages: strings, longs, lists, linkedLists, sets, maps, queues, etc.

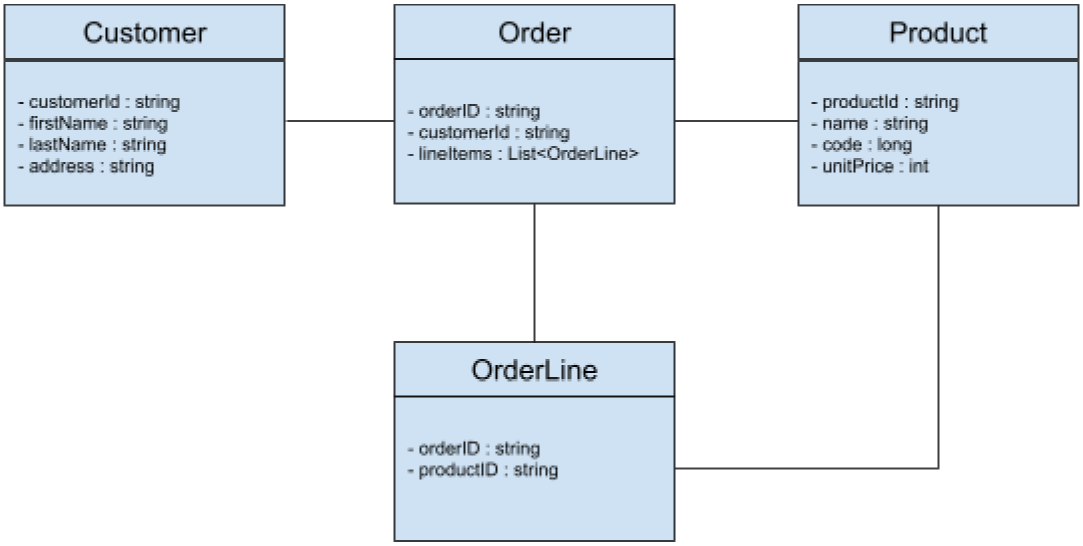


Figure 3: Class model example with members

Describe the relationship algorithm->model in the simple happy path scenario for two of ovens with three of the arms