

In this document, the specification for Festival Foods company is given using sets. Remember that Festival Foods is the company that buys from Florida Fruits and the specification shows the transactions performed by Festival Foods.

The state of the system consists of two sets, one for each type of fruits. The size of a set gives the number of fruits in that type. The state also consists of the buying price for each type of fruit. The state invariant asserts that the price of each fruit must be a positive number.

Initially, the two sets are empty and the price is set to some value.

The operation *UpdateOranges* updates the set *oranges* to reflect the purchase of oranges from Florida Fruits. This operation also outputs the cost of the purchase using the buying price for oranges. There is a similar operation for plums. There are two operations, one for each type of fruit to change the price of that type of fruit.

[*ORANGE*, *PLUM*]

FestivalFoods

oranges : \mathbb{P} *ORANGE*
plums : \mathbb{P} *PLUM*
priceForOrange : \mathbb{R}
priceForPlum : \mathbb{R}

priceForOrange > 0.0
priceForPlum > 0.0

INIT

oranges = \emptyset
plums = \emptyset
priceForOrange = 1.19
priceForPlum = 0.59

UpdateOranges

Δ (*oranges*)
howMuch? : \mathbb{P} *ORANGE*
cost! : \mathbb{R}

$\#$ *howMuch?* > 0
oranges' = *oranges* \cup *howMuch?*
cost! = ($\#$ *howMuch?*) * *priceForOrange*

ChangePriceForOrange

Δ (*priceForOrange*)
newPrice? : \mathbb{R}

newPrice? > 0.0
priceForOrange' = *newPrice?*

UpdatePlums

Δ (*plums*)
howMuch? : \mathbb{P} *PLUM*
cost! : \mathbb{R}

$\#$ *howMuch?* > 0
plums' = *plums* \cup *howMuch?*
cost! = ($\#$ *howMuch?*) * *priceForPlum*

ChangePriceForPlum

Δ (*priceForPlum*)
newPrice? : \mathbb{R}

newPrice? > 0.0
priceForPlum' = *newPrice?*