* Products
  + Apple
  + Orange
  + etc
* Product Properties
  + Has individual Price
  + Pricing Rules
    - Can have multi-price rules
    - Buy 1 get 1 free
    - Buy 1 get 1 half price
    - Buy n item for y money
  + Rules changes frequently; can be updated at the start of checkout
* Checkout
  + Accepts items in any order (i.e apple, orange, orange, apple….)
  + applies Product Rules based on the total number and adjust price accordingly
* Console Application:
  + Takes input as an unordered list of singular items from shopping cart
  + Checks them out
  + Print itemized receipt
  + Print total price

# Solution

Product

* name
* id
  + has\_many pricingRules

Order

* + has\_many checkedOutItems

CheckOutItem

* + product
  + quantity

PricingRule

* + has\_many successors

* + name
  + quantity
  + price

* + calculate(quantity)

returns totalPrice after applying rule, remainingQuantity, total itmes ruleAppliedOn

PriceCalculator

calculate\_prices(order)

1. find total number for individual product
2. for each product
   1. apply pricing rule recursively to find our products price
3. find total price combining individual products price
4. Generate itemized receipt

Sample dataset:

#products

apple = Product("Apple", "A")

orange = Product("Orange", "O")

pear = Product("Pear", "P")

strawberry = Product("Strawberry", "S")

# product rules

# rules for Apple

apple.pricingRules = [

PricingRule(name = "Buy 3 for $1.3", quantity = 3, price = 1.3),

PricingRule(name = "Buy 1 for $0.5", quantity = 1, price = 0.5)

]

# rules for orange

orange.pricingRules = [

PricingRule(name = "Buy 1 get 1 half", quantity = 2, price = 1.5),

PricingRule(name = "Buy 1 for $1", quantity = 1, price = 1.0)

]

# rules for pear

pear.pricingRules = [

PricingRule(name = "Buy 1 get 1 free", quantity = 2, price = 0.5),

PricingRule(name = "Buy 1 for $0.5", quantity = 1, price = 0.5)

]

# rule for Strawberry

strawberry.pricingRules = [

PricingRule(name = "Buy 1 pack for $5", quantity = 1, price = 5.0)

]

# unsorted checked out items

# feel free to change the order, add, remove items

order = Order(checkedOutItems = [

CheckedOutItem(product = apple),

CheckedOutItem(product = orange),

CheckedOutItem(product = apple),

CheckedOutItem(product = pear),

CheckedOutItem(product = orange),

CheckedOutItem(product = strawberry),

CheckedOutItem(product = apple),

CheckedOutItem(product = apple),

CheckedOutItem(product = pear),

CheckedOutItem(product = orange)

]

)