Problem 2: Supermarket Pricing

After thinking of many way to solve this problem I have decided on one solution that I feel most comfortable with.

As given by the original problem I will use the base(regular) cost of $0.65 for a can(one unit) of beans as my example where possible. To solve this problem I decided that I would need the following classes to start with :
Product Group, Product and Pricing Options

So using the example, the Product Group would be "Canned Goods" that would contain a list of Products of witch "Beans" would be one. The Product "Beans" would then be link to a variety of Pricing Options and consist of : Product Name, Cost, Amount (with will be used by the Pricing Options and retrieved from the inventory),Price(which would be determined by the Pricing Option and returned).

Three for a dollar (so what's the price if I buy 4, or 5?)

"Tree for a dollar" would be a new Pricing Option that will be implemented for a product with a Amount of 3(units) or greater. So given 3 cans of "Beans" will cost $1 the Pricing Option will use the Amount of that product in the inventory and see how many time three divides into the amount then allocate $ 1 to the price for each time and put aside what's left to be calculated by using the regular price. Thus making a purchase of 4 cans will be $1 from the first 3 cans and the one can that is left will receive the regular price of $0.65 making the total price $1.65. Purchasing 5 cans will be thee cans for $1 and two cans timed the regular price of $0.65 making the total $2.3. This Pricing Option would then return the total as the Price to the Product.

$1.99/pound (so what does 4 ounces cost?)

Now showing that the Product has weight that determines the selling price I would create a new class called Weigh-Products that will consist of Weight and Unit properties. Here I would work in price compared to units. So using Apples as a example and that $1.99 per pound thus 1 pound is one Unit, using this we can determine the amount of units by using weight it the Pricing Option. So to answer, there's 16 ounces in a pound so four ounces is 1 fourth of a pound thus giving us a fourth of a unit. The total then would be $1.99 divide by four and return a total price of $0.4975. But because we are working with money the fractional total of 0.4975 will be rounded to $0.50.

Buy two; get one free (so does the third item have a price?)

I this instance the Pricing Option will be applied to the Product as well as the Inventory(Cart). This Pricing Schema will affect every can in the Inventory setting every third cans price to $0. To answer the question I would say the third item would still have the same cost but the price that it is sold for is set to 0 so the third item has a sale price $0.

Does fractional money exist?

I my opinion yes fractional money does exist.

When (if ever) does rounding take place?

Rounding would take place when working with fractional money. It will always be rounded to the nearest cent so the price will always be rounded to 4 decimals, but shown(bought) at 2 decimals which is a whole cent .

How do you keep an audit trail of pricing decisions (and do you need to)?

Writing a pricing decision to a log or data bases when the Pricing Option is applied would be a good idea. In the case of price changes and profit reviews I would say it's needed to keep an audit trail.

Are costs and prices the same class of thing?

Yes, sins in this instance the cost of the product is also the regular price.

If a shelf of 100 cans is priced using "buy two, get one free", how do you value the stock?

Here I would apply a Pricing Option to the cost of each can(product) in the Product Group. That would the calculate the value of the stock on the shelf.

Written By : Ruan Fourie