Coding Exercise

v. 1.1.3

Overview

GroceryCo wants to develop a kiosk checkout system for customers in their supermarket stores. The client wants a system that allows customers perform a “checkout” based on prices and promotions defined by GroceryCo. They plan on installing kiosks in their stores as a supplement to cashiers.

Implement a prototype checkout system which handles their pricing schemes.

The goal of this exercise is to demonstrate your attention to detail and thinking from a design perspective. Create a solution that would be worthy to put into production and can be extended by other members of the team.

Details

Write a console program in C#, which takes as input a simple file containing an unsorted list of items which represent a product from the customer’s basket scanned at the kiosk.
For example:

Apple
Orange
Banana

The program should perform a “check-out” process:

- Determine the effective price of each item
- Calculate and display the final total due from the customer

The checkout process should be able to accept the same item multiple times and in any order.
For example, an Apple might be scanned, followed by two Oranges, and then an Apple again.

Prices for GroceryCo’s products are defined individually in their current price catalog.
For example, 'Apple' might cost $0.75, or ‘Bananas’ cost $1.

Products will occasionally be eligible for promotions for a limited time as defined by the marketing team.

Support the following promotion:

- **On Sale price.**
  Selected products are purchased at a discounted price which is less than regular price.
  For example, 'Apple': $0.50 Sale Price.

The marketing team should be able to apply promotions to any product.
In the future, they may come up with other promotion types as well (see *Advanced Requirements*)

At the end of the checkout, print an itemized receipt containing the regular price for each item, discount or saving applied for a qualifying promotion, and the total price to be paid by the customer for the basket.

Prices and promotions often change at the last minute. Each time a checkout transaction begins, the system should utilize the latest pricing rules (both for regular prices and promotions). These can be read in from files. The format describing regular prices and promotions can be at the discretion of the developer but should be accessible for GroceryCo staff (product supply, marketing).

Advanced Requirement(s):
Support the following promotions:

- **Group promotional price based on the quantity purchased.**
  Products purchased which reach a specified quantity have a discounted price.
  For example, Buy 3 'Apple' for $2.00.
- **Additional product discount.**
  Products purchased which reach a specified quantity will discount an additional product of the same kind. For example, “buy one get one free” or “buy one, get one for 50% off”

Submission

Submit your solution via a GitHub repository.
Include a README documentation with any necessary information (notes, limitations/design choices, or assumptions).
The solution may use third party libraries – these must be referenced by NuGet.
Submissions are due one week or (7) days from receipt of this exercise.

Criteria

The solution will be examined from these perspectives:

- **Design**
  - Does the code demonstrate good object oriented principles?
- **Readability**
  - Is the code readable to someone familiar with C#?
- **Maintenance**
  - How much developer effort is needed to extend and maintain the code?
- **Testing**
  - How well tested is the code?
- **Operational-ness**
  - Does the solution work correctly and efficiently?
- **Aesthetics**
  - Is the code elegant and nice to read?