

Разработать консольное приложение, реализующее функционал формирования чека в магазине.

1. В данном задании важно показать понимание ООП, умение строить модели (выделять классы, интерфейсы, их связи), разделять функционал между ними, а также знать синтаксис самого языка. Обязательно применение паттернов проектирования (Factory, Builder, Decorator). Обратить внимание на устойчивость к изменениям в логике и избегать копипаста.
2. Использовать Java 17, gradle 7.5.
3. Приложение запускается `java RunnerClassName <набор_параметров>`, где набор параметров в формате `itemId-quantity` (`itemId` - идентификатор товара, `quantity` - его количество).
Например: **java CheckRunner 3-1 2-5 5-1 card-1234** должен сформировать и вывести в консоль чек содержащий в себе наименование товара с `id=3` в количестве 1шт, то же самое с `id=2` в количестве 5 штук, `id=5` - одна штука и т. д. **Card-1234** означает, что была предъявлена скидочная карта с номером 1234. Необходимо вывести в консоль сформированный чек (вариант на рисунке), содержащий в себе список товаров и их количество с ценой, а также рассчитанную сумму с учетом скидки по предъявленной карте (если она есть).
4. Среди товаров предусмотреть акционные. Если их в чеке больше пяти, то сделать скидку 10% по этой позиции. Данную информацию отразить в чеке.
5. Набор товаров и скидочных карт может задаваться прямо в коде, массивом или коллекцией объектов. Их количество и номенклатура имеет тестовый характер, поэтому наименование и количество свободные.
6. Реализовать обработку исключений (например, товара с `id` или файла не существует и т. д.).
7. Реализовать вывод чека в файл.
8. Описать README.md файл (используемый стек, инструкцию по запуску, описание эндпоинтов, если есть).
9. Использовать сборщик проекта gradle.
10. Исходники разместить в любом из публичных репозиториях (bitbucket, github, gitlab).
11. Организовать чтение исходных данных (товары и скидочные карты, формат тот же) из файлов (в таком случае можно передавать имя файла в набор параметров команды **java**).
12. Покрыть функционал юнит-тестами (не менее 70%).
13. * Замена хранения исходных данных (п.11) в файлах на PostgreSQL; сделать 2 таблицы (`product` и `discount_card`); DDL операции должны храниться в `src/main/resources` в файле с расширением `*.sql`; настройки подключения к БД хранить в `application.properties`.
14. ** Реализовать RESTFUL - интерфейс (например, получать чек по GET <http://localhost:8080/check?itemId=1&itemId=1>). Реализация на выбор (Servlet, Spring). UI реализовывать не нужно.

15. *** Развернуть приложение и PostgreSQL в Docker
16. **** Расширить функционал на свое усмотрение.

“*” - Необязательно, но будет существенным плюсом. К этим пунктам лучше приступать после качественного решения базовых задач с применением принципов SOLID, декларативных подходов, оптимальных алгоритмов.

CASH RECEIPT			
SUPERMARKET 123			
12, MILKYWAY Galaxy/ Earth			
Tel :123-456-7890			
CASHIER: #1520		DATE: 12/12/2023	
		TIME: 23:24:18	
QTY	DESCRIPTION	PRICE	TOTAL
1	Lorem ipsum	\$1,55	\$1,55
1	Dolor	\$2,34	\$2,34
3	Sir amet	\$3,32	\$9,96
2	Consectetur adipin	\$10,50	\$21,00
1	Elit	\$3,12	\$3,12
1	Suspendisse eget	\$0,45	\$0,45
3	Placerat massa	\$37,34	\$112,02
1	Aenean vulputate	\$17,43	\$17,43
2	Quam ac eleifend	\$1,50	\$3,00
1	Pharetra	\$0,99	\$0,99
4	Integer	\$4,23	\$12,92
1	Magna in	\$32,45	\$32,45
1	Lorem scelerisque	\$54,30	\$54,30
2	Efficitur	\$2,25	\$5,50
2	Aliquam	\$3,50	\$7,00
3	Erat volutpat	\$18,50	\$55,50
1	Etiam bibendum	\$9,75	\$9,75
6	Mauris mauris	\$0,65	\$3,90
1	Eget eleifend	\$1,75	\$1,75
5	Justo pulvinar	\$10,12	\$50,60
1	Quisque	\$8,17	\$8,17
2	Ullamcorper	\$5,63	\$11,26
1	At velit	\$17,80	\$17,80
3	In feugiat	\$9,99	\$29,97
2	Aliquet venenatis	\$7,55	\$15,10
3	Nisi id	\$0,45	\$1,35
2	Malesuada vestibui	\$10,35	\$20,70
1	Sicet lacus	\$53,70	\$53,70
1	Nunc tempor	\$0,25	\$0,25
TAXABLE TOT.		\$541,27	
VAT17%		\$22,64	
TOTAL		\$563,91	