

ДОКЛАД

Никола Тодоров

Курс: СИТВ408 Програмиране с Java

Преподавател: гл. ас. д-р Христина Костадинова

1. Общ преглед:

Целта на проекта е да се моделира работата на магазин – зареждане и продажба на стоки, изчисляване на финансови резултати и издаване на касови бележки. Проектът е изграден на базата на обектно-ориентирано програмиране с използване на изключения, сериализация и файлов вход/изход.

2. Класове и тяхната роля:

- Product: модел на една стока, включваща ID, име, цена, категория, срок на годност и количество. Метод за изчисление на крайна цена с надценка и отстъпка.
- Cashier: модел на касиер, който има име, ID и заплата.
- Receipt: модел на касова бележка. Съдържа номер, дата, списък от продукти, сума. Записва се в .txt и .ser файл.
- Store: съдържа списъци с наличните продукти, касиери и касови бележки. Изчислява приходи, разходи и печалба. Метод за обработка на покупка и издаване на бележка.

3. Логика на продажба:

При подаване на заявка за покупка се проверява дали продуктът съществува и дали срокът му на годност не е изтекъл. Ако всичко е наред, се проверява дали има достатъчно количество. Ако не – се хвърля изключение. Продажната цена се пресмята в зависимост от категорията, надценката и дали остава малко време до изтичане на срока. Генерира се обща сума и се издава касова бележка, която се записва в .txt и .ser файл.

4. Работа с файлове:

При всяка покупка се създават два файла:

- .txt файл с информация за покупката, която може да се чете от потребителя
- .ser файл със сериализирания обект Receipt, който може да бъде зареден повторно с Java

5. Обработка на изключения:

Класът `InsufficientQuantityException` се използва когато липсва нужното количество от дадена стока. Съобщението показва с точност колко броя не достигат.

6. Изчисления на финанси:

Разходите включват заплатите на касиерите и цената за доставка на стоките. Приходите са базирани на стойността на всички издадени касови бележки. Печалбата се изчислява като разлика между приходите и разходите.

7. Демонстрация в `main` метода:

Създава се обект `Store`. Добавят се касиер и продукти. Извършва се покупка, която генерира касова бележка. Показва се оборот, разход и печалба. Използва се и метод за четене на съхранена бележка от файл.

8. JUnit тестове:

JUnit тестове не са включени, за да се запази кодът максимално опростен. При желание могат да се добавят в бъдеще.

9. Организация на проекта:

Проектът е реализиран в единен Java файл, с всички класове подредени в логичен ред. Това улеснява четенето и представянето на кода. Няма нужда от външни библиотеки.