

Невідомо, що це за магазин(1), для кого ця система(2), яка її задача(3). Через всі ці завдання проходить тільки одне слово “магазин”, що само по собі означає, виключно те, що це якось пов'язано із продажами.

- 1) Якщо це спеціалізований магазин (продаж товарів, пов'язаних однаковою тематикою. Наприклад: музика, риболовля, спорт, одяг, меблі і тд.), то створення ієрархічної системи класів та інтерфейсів є, на мою думку, дуже доречною тому, що таким чином можливо визначити кордони відповідальності і можливостей програми (зробити її кінечною, “невсесильною”), створити специфічну до тематики бізнес логіку.

Якщо це рітейл магазин, то такий підхід, на мою думку, може дуже швидко перетворити його розробку у жах:

- Продаж нової категорії товарів можливий виключно після того, як програмісти придумали новий функціонал та успішно протестували його. (Негнучкість системи у рітейл магазині!)
Така залежність від програмістів та постійний стан розробки може не сподобатися замовнику.
- Постійне створення нових сутностей може роздути програму (за рахунок одних тільки інтерфейсів та класів, без бізнес логіки) і зробити її підтримку і розробку дуже складною.

Для рітейл магазину, на мою думку, більше всього підходить гнучка система, в якій інформація про товар знаходиться у текстовому полі, а сам товар належить до якоїсь категорії (можливо створити ієрархію категорій для цілей автоматизації).

- 2) На мою думку, сутність “магазин” можна поділити 3 системи:
1. “Фронтенд” - де відбувається продаж товарів (сайт)
 2. Для працівників продажу - де продавці приймають та закуповують товари, роблять облік та спеціальні пропозиції і тд.
 3. Для працівників складу - де працівники складу приймають і відправляють товари, роблять облік та підтримують якість товару.

(3) Кожна із систем має за ціль автоматизувати відповідні задачі.

Якщо ця система репрезентує сайт інтернет-магазину (що, неоголошено але логічно, виходить із завдання 14), то:

- Створення ієрархії сутностей має зміст (наприклад, для фільтрації по категоріям), але функціонал терміну придатності є проблематичним і є, на мою думку, відповідальність складської бізнес логіки.
- (3) Навіщо у цій “фронтенд” системі складська логіка, якщо у інтернет-магазину може бути декілька складів*, деякі з яких можуть бути з незалежним до магазину керівництвом?
- *Навіщо реалізовувати паттерн Singleton для класу Storage?

- Якщо на сайті буде функціонал перегляду наявності товару на певному складі, то чи не варто зробити API складської системи та сайту?