

## Задание

Реализация системы управления инвентарем в магазине

### Описание

Вам предстоит разработать систему управления инвентарем в магазине на языке Ruby. Система должна позволять добавлять товары, управлять их количеством, отслеживать заказы и продажи. Для выполнения задания используйте классы, методы и базовые конструкции Ruby.

### Технические требования

#### Класс Product

Атрибуты:

name (строка) — название товара

sku (строка) — уникальный идентификатор товара (артикул)

price (число) — цена товара

quantity (число) — количество товара на складе

Методы:

initialize(name, sku, price, quantity) — конструктор, устанавливающий начальные значения атрибутов.

to\_s — метод для отображения информации о товаре в виде строки.

restock(amount) — метод для пополнения запасов товара.

sell(amount) — метод для продажи товара (уменьшает количество на складе).

#### Класс Order

Атрибуты:

order\_id (строка) — уникальный идентификатор заказа

products (хэш) — хэш, где ключи — объекты Product, а значения — количество заказанных товаров

total\_price (число) — общая стоимость заказа

Методы:

initialize(order\_id) — конструктор, устанавливающий начальные значения атрибутов, products инициализируется как пустой хэш, total\_price как 0.

add\_product(product, quantity) — метод для добавления товара в заказ.

to\_s — метод для отображения информации о заказе в виде строки.

#### Класс Store

Атрибуты:

products (массив) — коллекция товаров в магазине

orders (массив) — коллекция заказов в магазине

Методы:

initialize — конструктор, инициализирующий пустые массивы для товаров и заказов.

add\_product(product) — метод для добавления товара в магазин.

find\_product(sku) — метод для поиска товара по артикулу.  
create\_order(order\_id) — метод для создания нового заказа.  
list\_products — метод для отображения списка товаров.  
list\_orders — метод для отображения списка заказов.

### Указания к выполнению

Создайте классы Product, Order и Store согласно техническим требованиям.  
Напишите методы для каждого класса, реализуя описанные функции.  
В основном скрипте создайте несколько товаров, добавьте их в магазин.  
Создайте несколько заказов, добавив в них товары и посчитайте общую стоимость заказа.  
Выведите списки товаров и заказов на экран.

### Пример использования

```
# Создание магазина
store = Store.new

# Создание товаров
product1 = Product.new("Laptop", "SKU001", 1000.0, 10)
product2 = Product.new("Phone", "SKU002", 500.0, 20)
product3 = Product.new("Tablet", "SKU003", 300.0, 15)

# Добавление товаров в магазин
store.add_product(product1)
store.add_product(product2)
store.add_product(product3)

# Создание заказа
order1 = store.create_order("ORDER001")
order1.add_product(product1, 2)
order1.add_product(product2, 1)

# Создание другого заказа
order2 = store.create_order("ORDER002")
order2.add_product(product2, 3)
order2.add_product(product3, 2)

# Вывод списков
puts "Products:"
store.list_products

puts "\nOrders:"
store.list_orders
```

### Критерии оценки

1. Правильность реализации классов и методов — насколько точно реализованы методы согласно требованиям.
2. Работоспособность — корректная работа всех функций (добавление товаров, создание заказов, продажа и пополнение товаров, вывод списков).
3. Чистота и структура кода — правильное использование отступов, имён переменных и методов, а также организация кода.
4. Комментарии и документация — наличие и качество комментариев, объясняющих работу методов и классов.