

1. Создать dataclass `Product`, который содержит в себе поля `id`, `name`, `price`. Указать для полей типы `int`, `str`, `float` соответственно.
2. Создать dataclass `Pizza` и `Coffee`, которые наследуются от `Product` и добавляют новые поля данных. Для пиццы это будет список начинки, острая или нет и диаметр. Для кофе будет объем стакана и его.
3. Создать абстрактный класс `AbstractShop`, который будет описывать абстрактные методы:

- `add\_product` — должен принимать тип `Product`
- `sell\_product` — должен принимать тип `Product`
- `all\_products` — должен возвращать список `Product`

Данный абстрактный класс описывает требуемые методы для работы магазина (добавление нового товара, продажа и перечень всех).

4. Создать класс `RealShop`, который будет наследоваться от `AbstractShop` и реализовывать все его абстрактные методы.

Данное заведение будет продавать пиццы и кофе.

Проверить работоспособность каждого метода.

5. Создать проверки типа передаваемого параметра для методов `add\_product` и `sell\_product`, на вход должны передаваться только продукты. Проверять тип необходимо через функцию `isinstance`. Если тип передаваемого значения не является, каким либо продуктом, то должна вызываться ошибка `NonProductError` (создать новый класс ошибки).
6. Создать магазин мебели, в котором будут продаваться столы, стулья и шкафы. Также необходимо создать на основе dataclass `Product` новый, общий для всех товаров в мебельном магазине, класс `Furniture`, каждый новый тип товара в этом магазине должен быть его предком.
7. Реализовать в методах `add\_product` и `sell\_product` проверку, что передаваемый продукт является мебелью через функцию `isinstance`. Если это не так, то необходимо вызвать ошибку `NonProductError`.