

4.1 Наследование в объектно-ориентированном программировании

6 out of 11 steps passed 11 out of 29 points received

Видео-разбор подвига (решение смотреть только после своей попытки): <https://youtu.be/K0w2oBTThAc> (<https://youtu.be/K0w2oBTThAc>)

Подвиг 5. Иногда наследование используют, чтобы наделить объекты дочерних классов определенным набором атрибутов. Сделаем такой пример.

Предположим, вы разрабатываете программу для интернет-магазина. В этом магазине могут быть как реальные (физические) товары, так и электронные. Для этих двух групп, очевидно, нужен разный набор атрибутов:

- для реальных физических товаров: `id`, `name`, `price`, `weight`, `dims`

где `id` - идентификатор товара (целое число); `name` - наименование товара (строка); `price` - цена товара (вещественное число); `weight` - вес товара (вещественное число); `dims` = (`length`, `width`, `depth`) - длина, ширина, глубина - габариты товара (вещественные числа);

- для электронных товаров: `id`, `name`, `price`, `memory`, `frm`

где `id` - идентификатор товара (целое число); `name` - наименование товара (строка); `price` - цена товара (вещественное число); `memory` - занимаемый размер (в байтах - целое число); `frm` - формат данных (строка: `pdf`, `docx` и т.п.)

Так как все товары могут идти вперемешку, то мы хотим, чтобы в каждом объекте (для товара) присутствовали все атрибуты:

`id`, `name`, `price`, `weight`, `dims`, `memory`, `frm`

с начальными значениями `None`. А уже, затем, нужным из них будут присвоены конкретные данные.

Для реализации этой логики объявите в программе базовый класс с именем `Thing` (вещь, предмет), объекты которого могут создаваться командой:

```
th = Thing(name, price)
```

А атрибут `id` должен формироваться автоматически и быть уникальным для каждого товара (например, можно для каждого нового объекта увеличивать на единицу).

В объектах класса `Thing` должен формироваться полный набор локальных атрибутов (`id`, `name`, `price`, `weight`, `dims`, `memory`, `frm`) со значением `None`, кроме атрибутов: `id`, `name`, `price`.

Далее, нужно объявить два дочерних класса:

`Table` - для столов;

`ElBook` - для электронных книг.

Объекты этих классов должны создаваться командами:

```
table = Table(name, price, weight, dims)
book = ElBook(name, price, memory, frm)
```

Причем, атрибуты `name`, `price` (а также `id`) следует инициализировать в базовом классе, т.к. они общие для всех товаров. Остальные атрибуты должны либо принимать значение `None`, если не используются, либо инициализироваться конкретными значениями уже в дочерних классах.

Наконец, в базовом классе `Thing` объявите метод:

`get_data()`. - для получения кортежа в формате (id, name, price, weight, dims, memory, frm)

Пример использования классов (эти строчки в программе писать не нужно):

```
table = Table("Круглый", 1024, 812.55, (700, 750, 700))
book = ElBook("Python 00П", 2000, 2048, 'pdf')
print(*table.get_data())
print(*book.get_data())
```

P.S. В программе нужно объявить только классы. Выводить на экран ничего не нужно.

To solve this problem please visit

<https://stepik.org/lesson/701995/step/6>