

```

1  """Урок 8 Задание 4, 5, 6
2  Начните работу над проектом «Склад оргтехники».
   Создайте класс, описывающий склад. А
3  также класс «Оргтехника», который будет базовым для
   классов-наследников. Эти классы –
4  конкретные типы оргтехники (принтер, сканер, ксерокс
   ). В базовом классе определить
5  параметры, общие для приведенных типов. В классах-
   наследниках реализовать параметры,
6  уникальные для каждого типа оргтехники.
7  Разработать методы, отвечающие за приём
8  оргтехники на склад и передачу в определенное
   подразделение компании. Для хранения
9  данных о наименовании и количестве единиц оргтехники
   , а также других данных, можно
10 использовать любую подходящую структуру, например
   словарь.
11 Реализуйте механизм валидации вводимых
12 пользователем данных. Например, для указания
   количества принтеров, отправленных на
13 склад, нельзя использовать строковый тип данных.
14 Подсказка: постарайтесь по возможности реализовать в
   проекте «Склад оргтехники»
15 максимум возможностей, изученных на уроках по ООП.
16 """
17
18 class StoreMachines:
19
20     def __init__(self, name, price, quantity,
   number_of_lists, *args):
21         self.name = name
22         self.price = price
23         self.quantity = quantity
24         self.numb = number_of_lists
25         self.my_store_full = []
26         self.my_store = []
27         self.my_unit = {'Модель устройства': self.
   name, 'Цена за ед': self.price, 'Количество': self.
   quantity}
28
29     def __str__(self):
30         return f'{self.name} цена {self.price}
   количество {self.quantity}'
31

```

```

32     def reception(self):
33
34         try:
35             unit = input(f'Введите наименование ')
36             unit_p = int(input(f'Введите цену за ед '
37 ))
38             unit_q = int(input(f'Введите количество '
39 ))
40             unique = {'Модель устройства': unit, '
41 Цена за ед': unit_p, 'Количество': unit_q}
42             self.my_unit.update(unique)
43             self.my_store.append(self.my_unit)
44             print(f'Текущий список -\n {self.my_store
45 }')
46         except:
47             return f'Ошибка ввода данных'
48
49         print(f'Для выхода - Q, продолжение - Enter')
50         q = input(f'---> ')
51         if q == 'Q' or q == 'q':
52             self.my_store_full.append(self.my_store)
53             print(f'Весь склад -\n {self.
54 my_store_full}')
55             return f'Выход'
56         else:
57             return StoreMachines.reception(self)
58
59
60 class Printer(StoreMachines):
61     def to_print(self):
62         return f'to print smth {self.numb} times'
63
64
65 class Scanner(StoreMachines):
66     def to_scan(self):
67         return f'to scan smth {self.numb} times'
68
69
70 class Copier(StoreMachines):
71     def to_copier(self):
72         return f'to copier smth {self.numb} times'
73
74
75 unit_1 = Printer('hp', 2000, 5, 10)

```

```
71 unit_2 = Scanner('Canon', 1200, 5, 10)
72 unit_3 = Copier('Xerox', 1500, 1, 15)
73 print(unit_1.reception())
74 print(unit_2.reception())
75 print(unit_3.reception())
76 print(unit_1.to_print())
77 print(unit_3.to_copier())
```