class technic:  
 clas = "Проводная"  
 warranty.period = "2 года"  
 brand = "Lenovo"  
 model = "GS123"  
 color = "Black"  
 def set(self, clas, color):  
 self.clas = clas  
 self.color= color  
class TV(technic):  
 def technic\_method(self, model, color):  
 self.model = model  
 self.color = color  
class notebook(technic):  
 def technic\_note(self, product, color):  
 self.product = product  
 self.color = color  
class monitor(technic):  
 def technic\_monitor(self, size, color):  
 self.size = size  
 self.color = color  
class printer(technic):  
 def technic\_printer(self, company, color):  
 self.company = company  
 self.color = color  
class mikrovolnovka(technic):  
 def technic\_method(self, temp):  
 self.temp = temp  
  
class employee:  
 def employee(self, position, color):  
 self.position = position  
 self.color = color  
  
class employee:  
 position = "employee"  
 salary = 123456  
 def eployee(self, position, salary):  
 self.position = position  
 self.salary = salary  
class kassir(employee):  
 position = "Кассир"  
class manager(employee):  
 position = "Менеджер"  
Manager = [  
 {'name': 'Nikita', 'position': "assistant"},  
 {'name': 'Andrey', 'position': "manager"},  
 {'name': 'Daniil', 'position': "Head master"}  
]  
Customer = [  
 {'name': 'Stepan'},  
 {'name': 'Konstantin'},  
 {'name': 'Lebron'}  
]  
Televisor = [  
 {'model': 'LG', 'price': 122354, 'quality': 42},  
 {'model': 'Samsung', 'price': 42564, 'quality': 13},  
 {'model': 'HP', 'price': 123548, 'quality': 22},  
]  
  
items = list()  
people = list()  
customer = list()  
  
  
# Добавить лист с телевизорами  
def create\_tele\_list(app\_items):  
 global items  
 items = app\_items  
  
  
def create\_item(model, price, quantity):  
 global items  
 items.append({'model': model, 'price': price, 'quality': quantity})  
  
  
# Найти телевизор модели module  
def read\_item(model):  
 global items  
 myitems = list(filter(lambda x: x['model'] == model, items))  
 if myitems:  
 return myitems[0]  
 else:  
 print("no")  
  
  
# Вывести список телевизоров  
def read\_items():  
 global items  
 return [item for item in items]  
  
def read\_names():  
 global items  
 return [item for item in items[0]]  
  
  
# Добавим список менеджеров  
def create\_Manager\_list(app\_items):  
 global people  
 people = app\_items  
  
# Добавим список покупателей  
def create\_Customer\_list(app\_items):  
 global customer  
 customer = app\_items  
  
  
# Вывести список покупателей  
def customer\_list():  
 global customer  
 return [item for item in customer]  
  
  
# Вывести список покупателей  
def meneger\_list():  
 global people  
 return [item for item in people]  
  
  
# Функция, которая будет убирать телефизор, который купили  
def buy\_televisor(model):  
 global items  
 idxs\_items = list(  
 filter(lambda i\_x: i\_x[1]['model'] == model, enumerate(items)))  
 if idxs\_items:  
 i, item\_to\_delete = idxs\_items[0][0], idxs\_items[0][1]  
 del items[i]  
  
def choice\_name(choice):  
 new\_model = input("Введите модель телевизора: ")  
 Oleg.find(new\_model)  
 if read\_item(new\_model):  
 ans = input("Собираетесь ли вы его приобретать? (Да/Нет): ")  
 if ans == "Да":  
 Oleg.buy("dexp")  
 print("Хорошего пользования ")  
  
class cust\_buy:  
  
 def \_\_init\_\_(self, name):  
 self.name = name  
  
 # Найти необходимый телевизор  
 def find(self, model):  
 self.model = model  
 if read\_item(model):  
 print("Он в наличии")  
 return True  
 else:  
 print("Выберите другой")  
  
 # купить телевизор  
 def buy(self, model):  
 self.model = model  
 if read\_item(model):  
 buy\_televisor(model)  
 print("Он работает")  
 else:  
 print("К сожалению, данная модель не доступна(")  
  
  
  
create\_tele\_list(Televisor)  
create\_Manager\_list(Manager)  
create\_Customer\_list(Customer)  
# create\_item("samsung", 15000, 15)  
print(read\_names())  
  
  
name = input("Введте имя пользователя: ")  
Oleg = cust\_buy(name)  
choice = int(input("Вы можете выбрать конкретную модель телевизора, "  
 "который хотите купить, либо посмотреть список всех телевизоров(1/2): "))  
if choice == 1:  
 choice\_name(choice)  
else:  
 read\_names()  
choice = int(input("Вы можете выбрать конкретную модель телевизора, "  
 "который хотите купить, либо посмотреть список всех телевизоров(1/2): "))  
if choice == 1:  
 choice\_name(choice)  
else:  
 read\_names