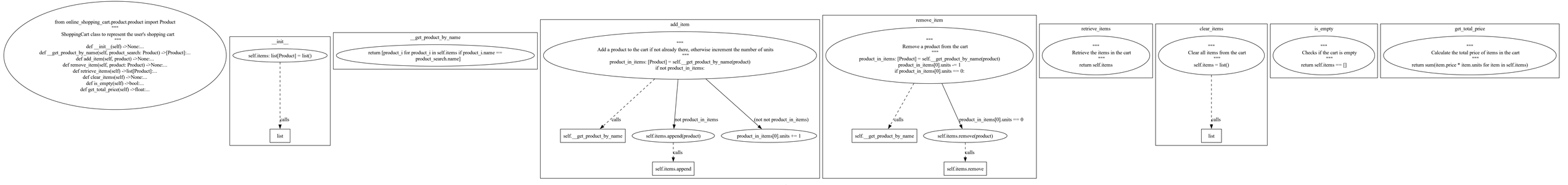
from online_shopping_cart.product.product import Product
"""
ShoppingCart class to represent the user's shopping cart
"""
def __init__(self) -> None:...
def __get_product_by_name(self, product_search: Product) -> [Product]:...
def add_item(self, product) -> None:...
def remove_item(self, product: Product) -> None:...
def retrieve_items(self) -> list[Product]:...
def clear_items(self) -> None:...
def is_empty(self) -> bool:...
def get_total_price(self) -> float:...
__init__
self.items: list[Product] = list()
calls
list
__get_product_by_name
return [product_i for product_i in self.items if product_i.name == product_search.name]
add_item
"""
Add a product to the cart if not already there, otherwise increment the number of units
"""
product_in_items: [Product] = self.__get_product_by_name(product)
if not product_in_items:
calls
self.__get_product_by_name
not product_in_items
self.items.append(product)
calls
self.items.append
(not not product_in_items)
product_in_items[0].units += 1
remove_item
"""
Remove a product from the cart
"""
product_in_items: [Product] = self.__get_product_by_name(product)
product_in_items[0].units -= 1
if product_in_items[0].units == 0:
calls
self.__get_product_by_name
product_in_items[0].units == 0
self.items.remove(product)
calls
self.items.remove
retrieve_items
"""
Retrieve the items in the cart
"""
return self.items
clear_items
"""
Clear all items from the cart
"""
self.items = list()
calls
list
is_empty
"""
Checks if the cart is empty
"""
return self.items == []
get_total_price
"""
Calculate the total price of items in the cart
"""
return sum(item.price * item.units for item in self.items)

cart