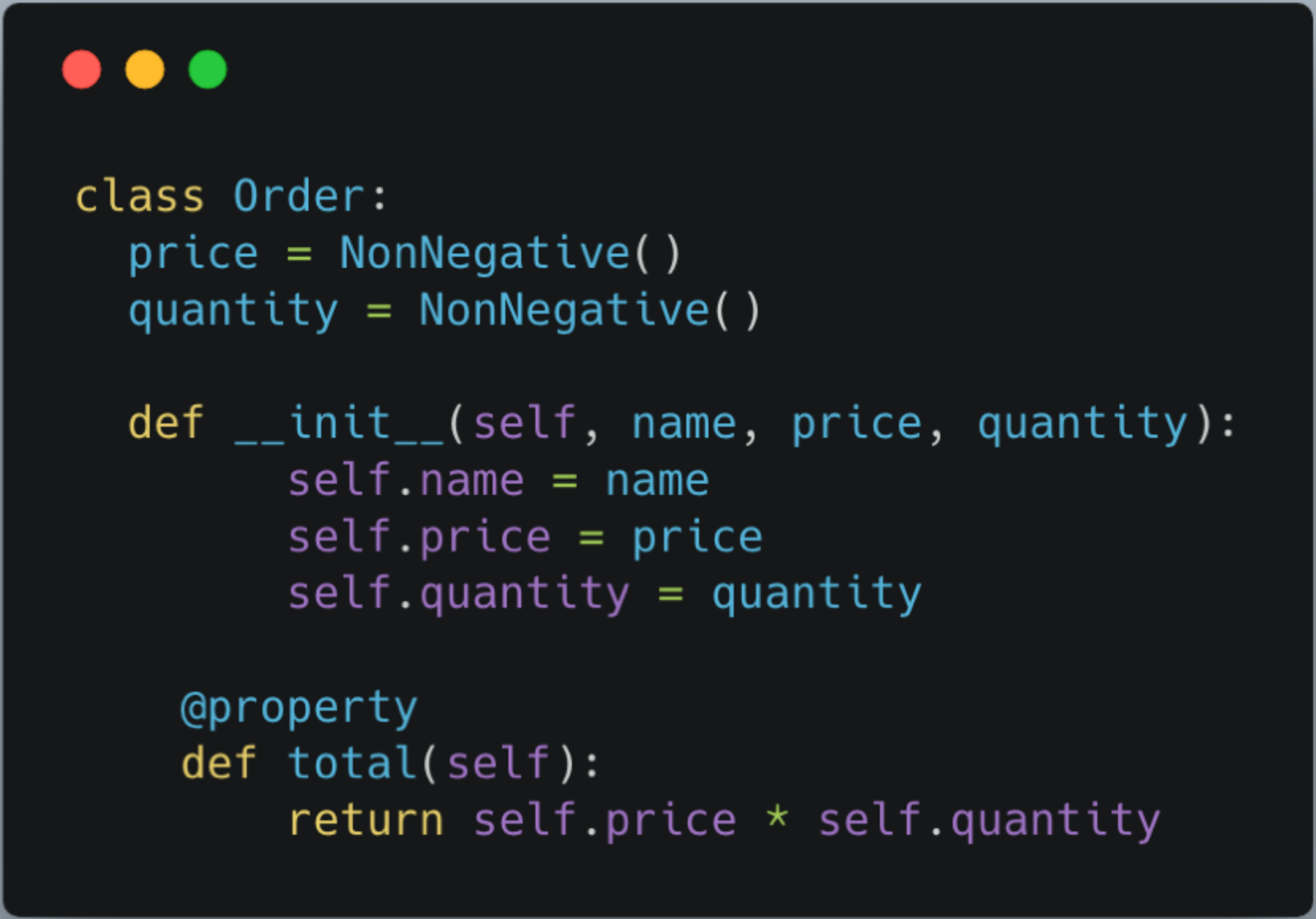
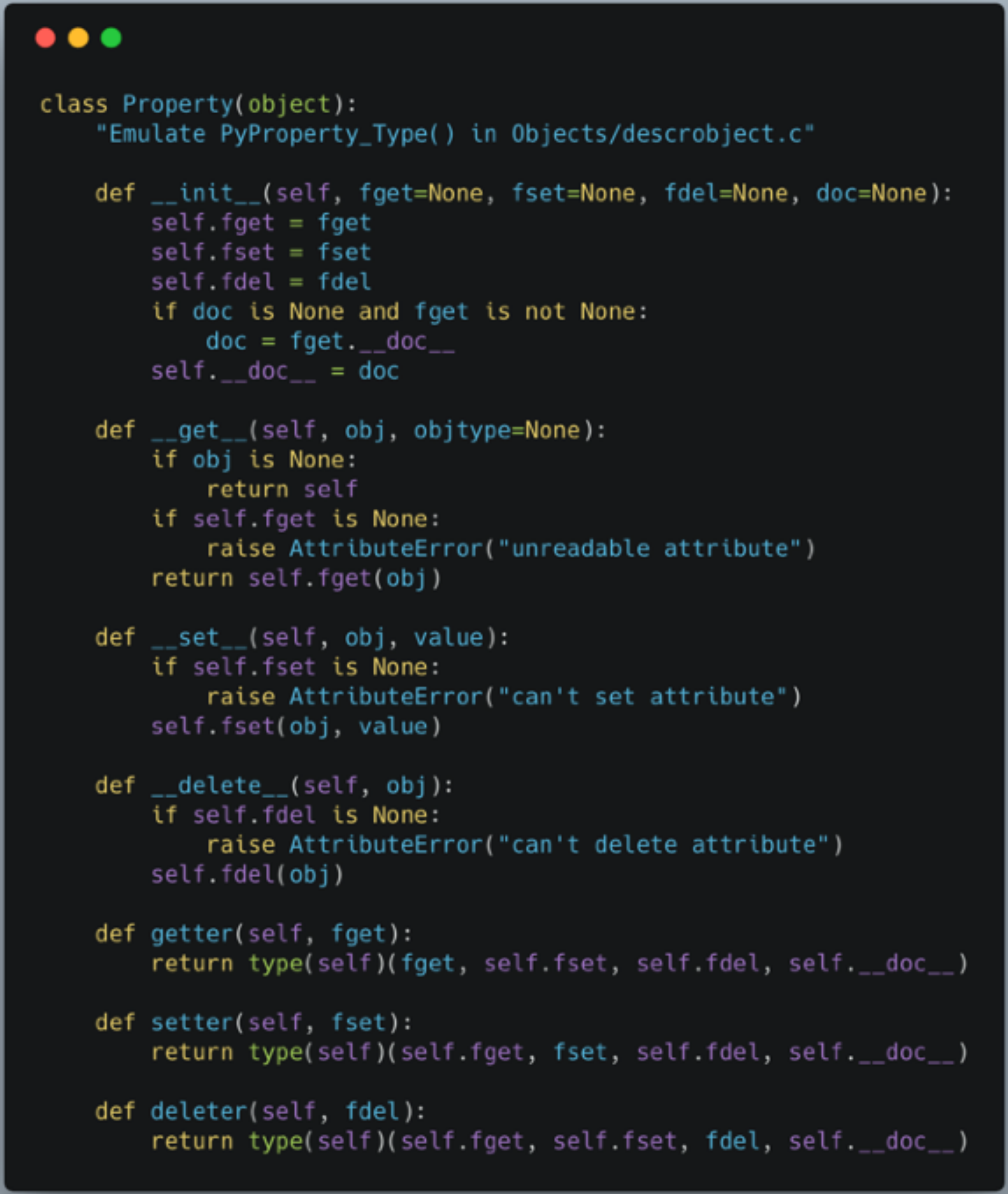
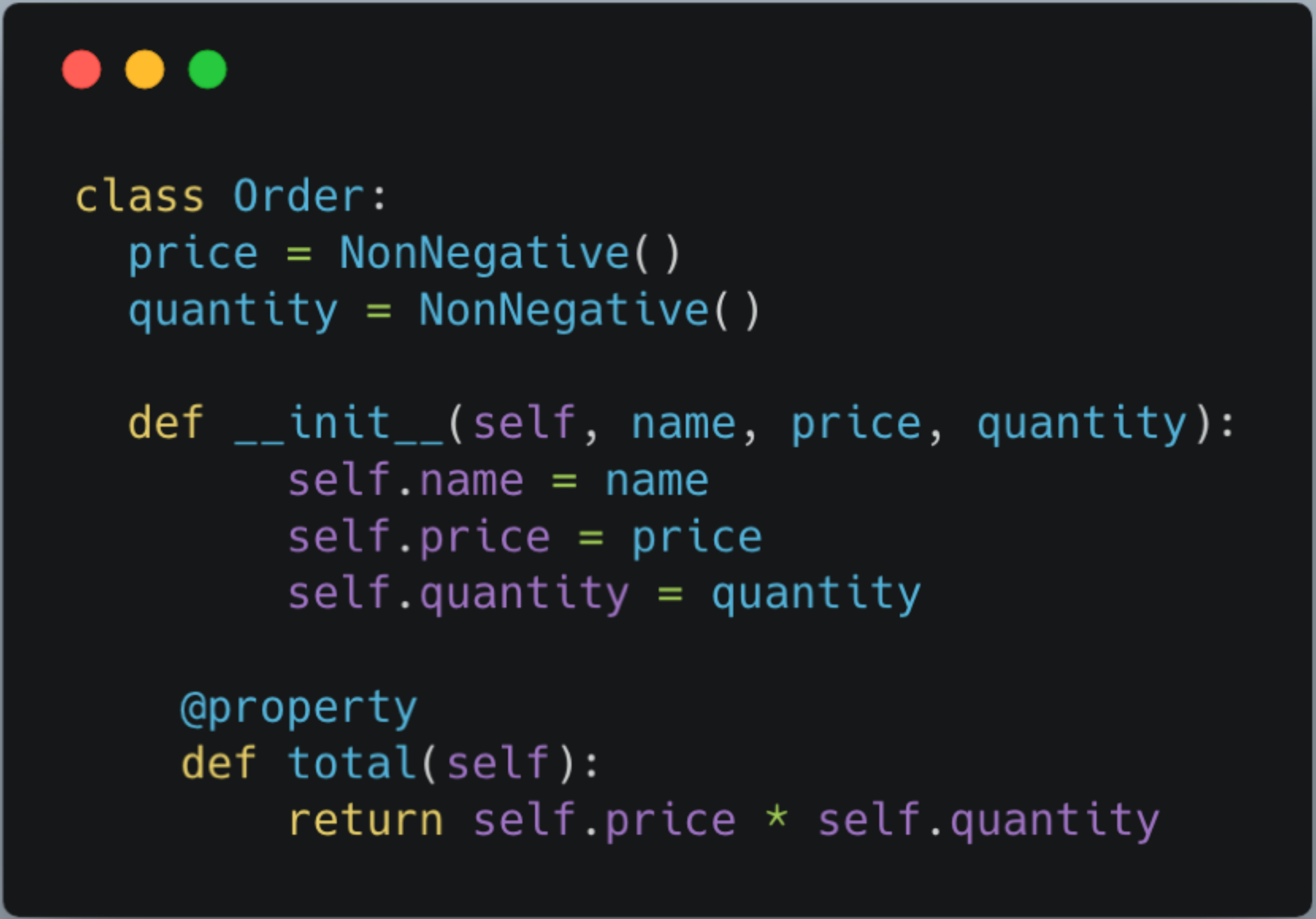


```
class Order:
  price = NonNegative()
  quantity = NonNegative()

  def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    @property
    def total(self):
        return self.price * self.quantity
```



```python
class Property(object):
    "Emulate PyProperty_Type() in Objects/descrobject.c"

    def __init__(self, fget=None, fset=None, fdel=None, doc=None):
        self.fget = fget
        self.fset = fset
        self.fdel = fdel
        if doc is None and fget is not None:
            doc = fget.__doc__
        self.__doc__ = doc

    def __get__(self, obj, objtype=None):
        if obj is None:
            return self
        if self.fget is None:
            raise AttributeError("unreadable attribute")
        return self.fget(obj)

    def __set__(self, obj, value):
        if self.fset is None:
            raise AttributeError("can't set attribute")
        self.fset(obj, value)

    def __delete__(self, obj):
        if self.fdel is None:
            raise AttributeError("can't delete attribute")
        self.fdel(obj)

    def getter(self, fget):
        return type(self)(fget, self.fset, self.fdel, self.__doc__)

    def setter(self, fset):
        return type(self)(self.fget, fset, self.fdel, self.__doc__)

    def deleter(self, fdel):
        return type(self)(self.fget, self.fset, fdel, self.__doc__)
```



```python
class Order:
  price = NonNegative()
  quantity = NonNegative()

  def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    @property
    def total(self):
        return self.price * self.quantity
```