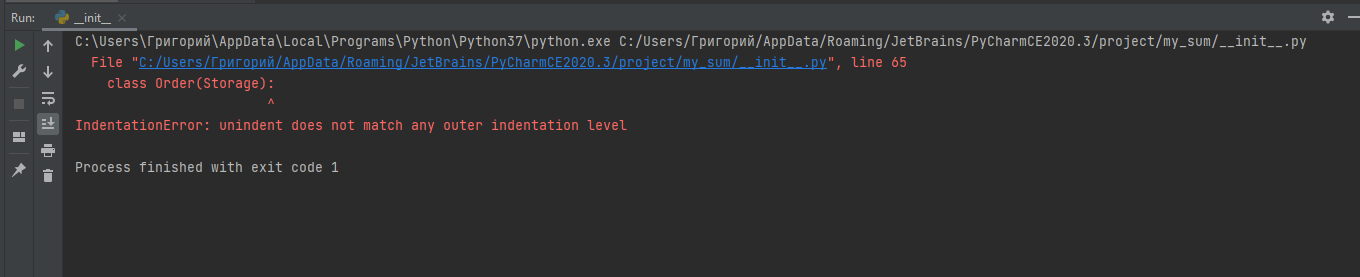
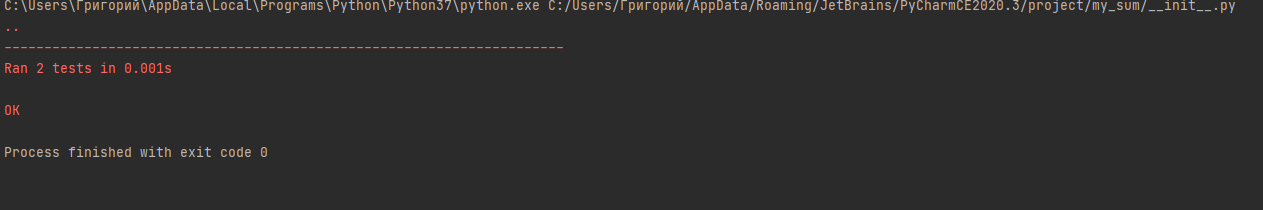
код программы до редактирования

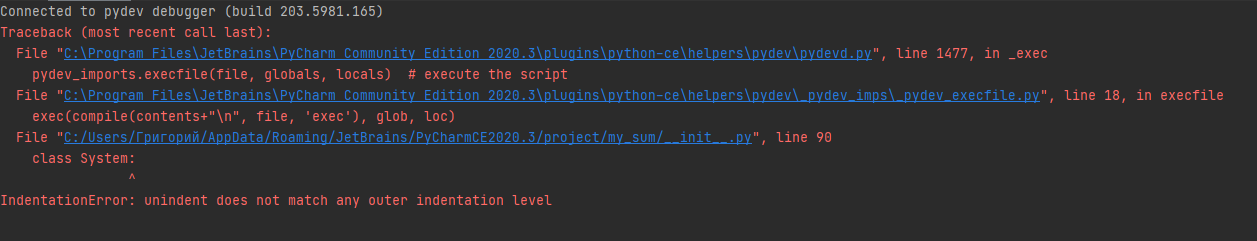
import unittest  
  
  
class Test(object):  
 *"""  
 >>> a=Test(1)  
 >>> a.multiply\_by\_2()  
 0.451  
 """* def \_\_init\_\_(self, weight):  
 self.weight = weight  
  
 def multiply\_by\_2(self):  
 return self.weight \* 0.451  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import doctest  
  
 doctest.testmod()  
  
  
class TestUM(unittest.TestCase):  
 def setUp(self):  
 pass  
  
 def tearDown(self):  
 pass  
  
 def test\_numbers\_3\_4(self):  
 self.assertEqual(3 \* 4, 12)  
  
 def test\_strings\_a\_3(self):  
 self.assertEqual('a' \* 3, 'aaa')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()  
  
  
class Storage:  
 def Storage\_method(self):  
 self.Storage = Storage  
  
  
class Goods(Storage):  
 *'''Система складского учета'''* def \_\_init\_\_(self, productname, weight, name):  
  
 self.productname = productname  
 self.weight = weight  
 self.count = count  
 self.dates = {}  
  
 def reserv(self, date, places):  
 reserved = instock  
 if places + reserved > self.places:  
 raise Exception  
 else:  
 instock = reserved + places  
  
  
class Order(Storage):  
 NEW = "New"  
 RESERVED = "Reserved"  
 COUNT = "Count"  
 ABSENT = "Absent"  
  
 def \_\_init\_\_(self, town, race, date, places, carrage):  
 self.productname = productname  
 self.weight = weight  
 self.count = count  
 self.date = date  
 self.places = places  
 self.state = Order.NEW  
  
 def reserv(self):  
 if self.state == Order.NEW:  
 try:  
 self.goods.reserv()  
 self.state = Order.RESERVED  
 except:  
 raise Exception  
 else:  
 raise Exception  
  
  
class System:  
 t1 = "Books"  
 t2 = "Sugar"  
 t3 = "Wine"  
  
 gc1 = Goods(t1, 1, 3)  
 gc2 = Goods(t2, 4, 1)  
 gc3 = Goods(t3, 5, 20)  
  
 orders = []  
  
 def order(self, productname, weight, count, date, places):  
 new\_order = Order(weight, productname, count, date, places)  
 self.orders.append(new\_order)  
  
 def reserv(self, order):  
 order.reserv()

Отчет, формируемый утилитой pydocstyle для исходного кода программы;  


отчет, формируемый утилитой pydocstyle для исправленного кода программы;



Отчет, формируемый утилитой pylint для исходного кода программы;



Отчет формируемый утилитой pylint для исправленного кода программы;

