Lab #4. Part 1

1. Modify the class Rational of Lab No2 to perform the following tasks:

- adding two Rational numbers. The result should be stored in reduced form;

- subtracting two Rational numbers. The result should be stored in reduced form;

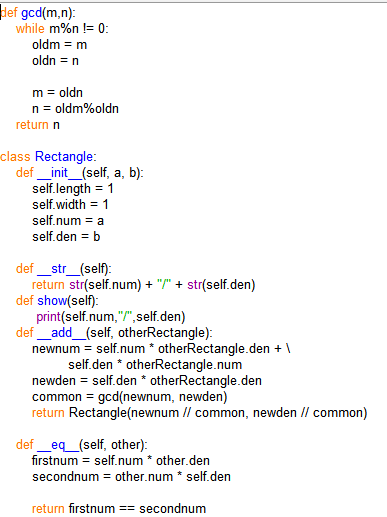
- multiplying two Rational numbers. The result should be stored in reduced form;

- dividing two Rational numbers. The result should be stored in reduced form;

- comparison two Rational numbers.

2. Create a class COMPOSITION with the names of goods, their quantity and price. Define methods for working with these data fields and overload operations to replenish product information, retrieve product information, generate a report on the availability of goods in stock according to the request.

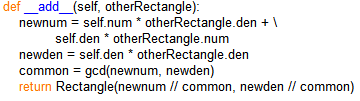
Ініціалізували клас



Функції класу

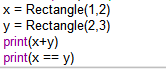
Виведення







Тест





Код

def gcd(m,n):

while m%n != 0:

oldm = m

oldn = n

m = oldn

n = oldm%oldn

return n

class Rectangle:

def \_\_init\_\_(self, a, b):

self.length = 1

self.width = 1

self.num = a

self.den = b

def \_\_str\_\_(self):

return str(self.num) + "/" + str(self.den)

def show(self):

print(self.num,"/",self.den)

def \_\_add\_\_(self, otherRectangle):

newnum = self.num \* otherRectangle.den + \

self.den \* otherRectangle.num

newden = self.den \* otherRectangle.den

common = gcd(newnum, newden)

return Rectangle(newnum // common, newden // common)

def \_\_eq\_\_(self, other):

firstnum = self.num \* other.den

secondnum = other.num \* self.den

return firstnum == secondnum

def per(self):

print(2 \* (self.length + self.width))

def sq(self):

print(self.width \* self.length)

def set\_length(self, length):

if length < 20.0:

self.length = length

def set\_width(self, width):

if width < 20.0:

self.width = width

def get\_width(self):

return self.width

def get\_length(self):

return self.length

def get\_ab(self):

print(f'числитель = {self.num} знаминатель = {self.den}')

print(f'{self.num / self.den}')

x = Rectangle(1,2)

y = Rectangle(2,3)

print(x+y)

print(x == y)