Lab #2. Part 1

1. Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide methods that calculate the perimeter and the area of the rectangle. Also, provide setter and getter for the length and width attributes. The setter should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0.2. Create a class called Rational for performing arithmetic with fractions. Write a program to test your class. Use integer variables to represent the private data of the class – the numerator and the denominator. Provide a \_\_init\_\_() method that enables an object of this class to be initialized when it’s declared. The \_\_init\_\_() should contain default parameter values in case no initializers are provided and should store the fraction in reduced form. For example, the fraction 2/4 would be stored in the object as 1 in the numerator and 2 in the denominator. Provide public methods that perform each of the following tasks:

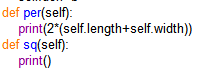
- printing Rational numbers in the form a/b, where a is the numerator and b is the denominator.

- printing Rational numbers in floating-point format.

Створили клас



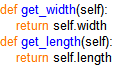
Ініціалізували 2 функції



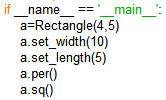
Задали сеттери



Гетери



Функція для тесту



Результат



Ініціалізували дріб



Вивід



Перевірка







**Part 2**

Lab #1. Part 2

Create a class that descibes a Product of online store. As a Product fields you can use a price, description and product' dimensions.

Create a class that describes a Customer. As a Customer fields you can use surname, name, patronymic, mobile phone, etc.

Create a class that describes an Order.

- the order must contain data about the customer who carried it out and products. Implement a method for calculating the total order value.Task 2.

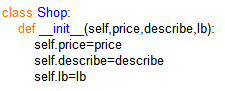
Create a class that performs statistical processing of a text file - counting characters, words, sentences, etc. Determine the required attributes-data and attributes-methods in class for working with the text file.Task 3.

The class GROUP contains a sequence of instances of the class STUDENT. The class STUDENT contains the student's name, surname, record book number and grades. Determine the required attributes-data and attributes-methods in classes GROUP and STUDENT. Find the average score of each student. Output to the standard output stream the five students with the highest average score.

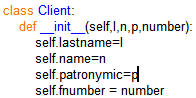
Assume that there can be no more than 20 students in a group, as well as students with the same name and surname.Task 4.

Create a class BINARY TREE that contains background information of product prices (product code, price of 1 product). The tree is sorted by product codes. From the keyboard enter data on the number of products in the format: product code, number of products. Using a tree, find the cost of products (cost = quantity \* price of one product).

Створили клас, який описує продукт інтернет-магазину. Як поля Товару можете використати ціну, опис та розміри товару.



Створіть клас, який описує клієнта. Як поля Замовника можна використовувати прізвище, ім'я, по батькові, мобільний телефон і т.д.

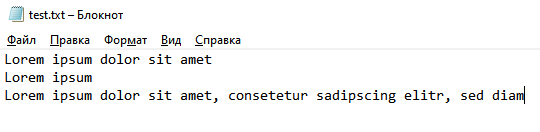


Клас із спадщиною для розрахунку загальної вартості

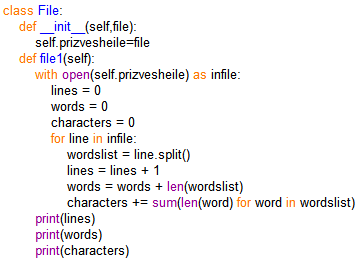


Створити клас, який виконує статистичну обробку текстового файлу – підрахунок символів, слів, речень тощо. Визначити в класі необхідні атрибути-дані та атрибути-методи для роботи з текстовим файлом

Створили тест файл



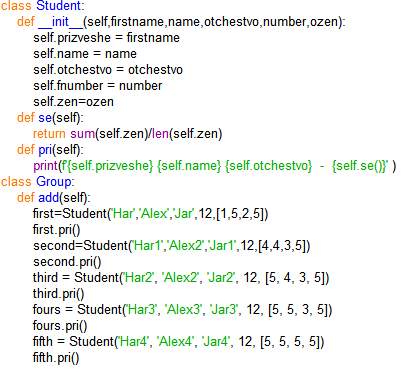
Клас



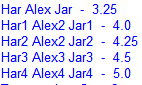




Клас GROUP містить послідовність екземплярів класу STUDENT. Клас СТУДЕНТ містить ім'я, прізвище учня, номер залікової книжки та оцінки. Визначити необхідні атрибути-дані та атрибути-методи у класах ГРУПА та СТУДЕНТ. Знайдіть середній бал кожного учня. Вивести у стандартний потік виведення п'ять студентів. Припустимо, що у групі може бути не більше 20 студентів, а також студентів з однаковим ім'ям та прізвищем.

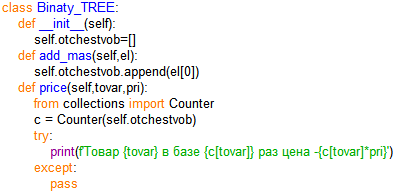






Створіть клас BINARY TREE, який містить довідкову інформацію про ціни на товари (код товару, ціна 1 товару). Дерево відсортовано за кодом товару. З клавіатури введіть дані про кількість товарів у форматі: код товару, кількість товарів. За допомогою дерева знайдіть собівартість продукції (собівартість = кількість \* ціна одного виробу).

Створили клас



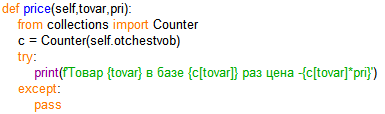
Ініціалізація загального масиву



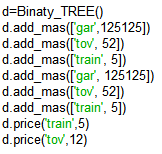
Додавання



Виведення потрібного товару за ціною



Тест





Код

class Rectangle:

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

self.length=1

self.width=1

self.num=a

self.den=b

def per(self):

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

def sq(self):

print(self.width\*self.length)

def set\_length(self,length):

self.length=length

def set\_width(self,width):

self.width=width

def get\_width(self):

return self.width

def get\_length(self):

return self.length

def get\_ab(self):

print(f'чисельник = {self.num} знаменник = {self.den}')

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

class Shop:

def \_\_init\_\_(self,price,describe,lb):

self.price=price

self.describe=describe

self.lb=lb

class Сlient:

def \_\_init\_\_(self,l,n,p,number):

self.lastname=l

self.name=n

self.patronymic=p

self.fnumber = number

class Order(Shop,Сlient):

def obsh(self):

return self.price

class File:

def \_\_init\_\_(self,file):

self.prizvesheile=file

def file1(self):

with open(self.prizvesheile) as infile:

lines = 0

words = 0

characters = 0

for line in infile:

wordslist = line.split()

lines = lines + 1

words = words + len(wordslist)

characters += sum(len(word) for word in wordslist)

print(lines)

print(words)

print(characters)

class Student:

def \_\_init\_\_(self,firstname,name,otchestvo,number,ozen):

self.prizveshe = firstname

self.name = name

self.otchestvo = otchestvo

self.fnumber = number

self.zen=ozen

def se(self):

return sum(self.zen)/len(self.zen)

def pri(self):

print(f'{self.prizveshe} {self.name} {self.otchestvo} - {self.se()}' )

class Group:

def add(self):

first=Student('Har','Alex','Jar',12,[1,5,2,5])

first.pri()

second=Student('Har1','Alex2','Jar1',12,[4,4,3,5])

second.pri()

third = Student('Har2', 'Alex2', 'Jar2', 12, [5, 4, 3, 5])

third.pri()

fours = Student('Har3', 'Alex3', 'Jar3', 12, [5, 5, 3, 5])

fours.pri()

fifth = Student('Har4', 'Alex4', 'Jar4', 12, [5, 5, 5, 5])

fifth.pri()

class Binaty\_TREE:

def \_\_init\_\_(self):

self.otchestvob=[]

def add\_mas(self,el):

self.otchestvob.append(el[0])

def price(self,tovar,pri):

from collections import Counter

c = Counter(self.otchestvob)

try:

print(f'Товар {tovar} в базе {c[tovar]} раз цена -{c[tovar]\*pri}')

except:

pass

if \_\_name\_\_ == '\_\_main\_\_':

a=Rectangle(4,5)

a.set\_width(10)

a.set\_length(5)

a.per()

a.sq()

a.get\_ab()

b=File('test.txt')

b.file1()

c=Group()

c.add()

d=Binaty\_TREE()

d.add\_mas(['gar',125125])

d.add\_mas(['tov', 52])

d.add\_mas(['train', 5])

d.add\_mas(['gar', 125125])

d.add\_mas(['tov', 52])

d.add\_mas(['train', 5])

d.price('train',5)

d.price('tov',12)