|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
|  |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт комплексной безопасности и специального приборостроения |
| Кафедра КБ-2 «Прикладные информационные технологии» |

**Общий отчет по всем лабораторным работам**

по дисциплине: **«Технологии и методы программирования»**.

Подготовил студент 3 курса группы БИСО-01-20 Полянская Полина Алексеевна

Студенческий: 20Б0866

Проверил преподаватель Выжигин Александр Юрьевич.

Содержание

**Лабораторная работа 13**

**Лабораторная работа 24**

**Лабораторная работа 37**

**Лабораторная работа 413**

**Лабораторная работа 517**

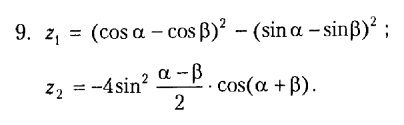
**Лабораторная работа 621**

**Лабораторная работа 938**

**Лабораторная работа №1**Общее задание, номер варианта, текст варианта

Напишите программу для расчета по двум формулам

Текст варианта №9:



Описание логической структуры программы

import math as mt

a = float(input("enter a = "))

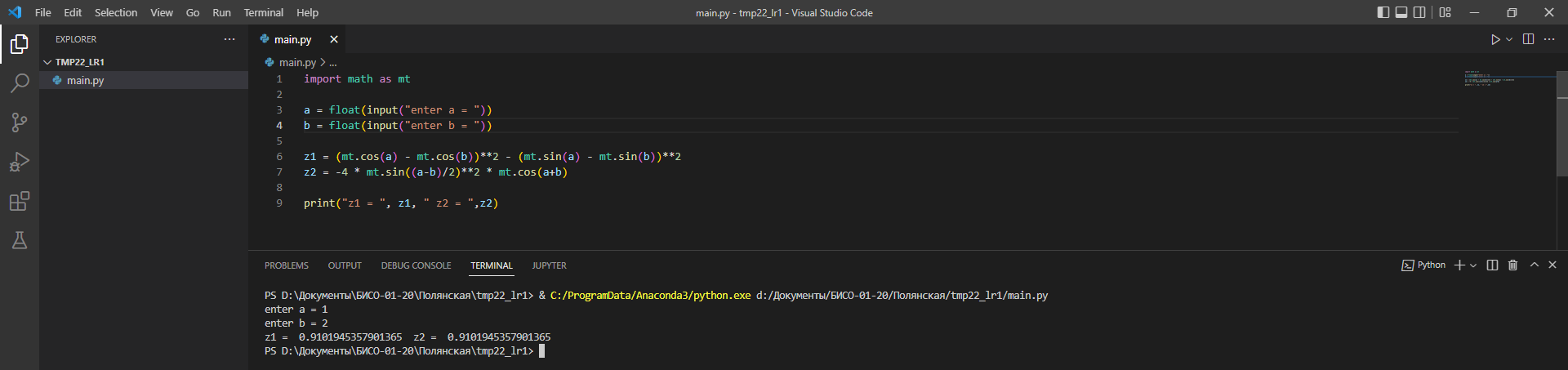
b = float(input("enter b = "))

z1 = (mt.cos(a) - mt.cos(b))\*\*2 - (mt.sin(a) - mt.sin(b))\*\*2

z2 = -4 \* mt.sin((a-b)/2)\*\*2 \* mt.cos(a+b)

print("z1 = ", z1, " z2 = ",z2)

Скриншоты работы программы



**Лабораторная работа №2**

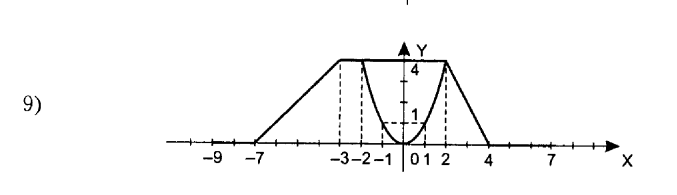
Общее задание

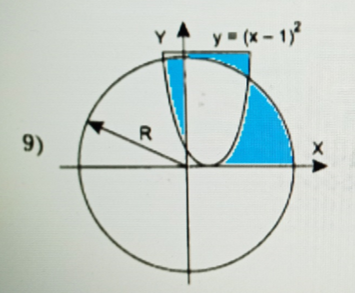
1. Напишите программу, которая по введённому значению аргумента вычисляет значение функции, заданной в виде графика. Параметр R вводится с клавиатуры
2. Написать программу, которая определяет, попадает ли точка с заданными координатами в область, закрашенную на рисунке серым цветом. Результат работы программы вывести в виде текстового сообщения.

Номер варианта, текст варианта

**Номер варианта 9**

Текст варианта:





Описание логической структуры программы

1) x = float(input("x = "))

if (x >= -9) and (x < -7):

print("y = ", 0)

elif (x >= -7) and (x < -3):

print("y = ", x+7)

elif (x >= -3) and (x < -2):

print("y = ", 4)

elif (x >= -2) and (x < 2):

print("y1 = 4, y2 = ", x\*\*2)

elif (x >= 2) and (x < 4):

print("y = ", -2\*x + 8)

elif (x >= 4) and (x <= 7):

print("y = 0")

else:

print("x не попадает в границы функции")

2)

r = 2

x = float(input("введите x = "))

y = float(input("введите y = "))

if (x\*x + y\*y <= r\*r) and (x <= 0) and (y >= (x-1)\*\*2) or \

(x\*x + y\*y >= r\*r) and (y<=r) and (x>=0) and (y >= (x-1)\*\*2) or \

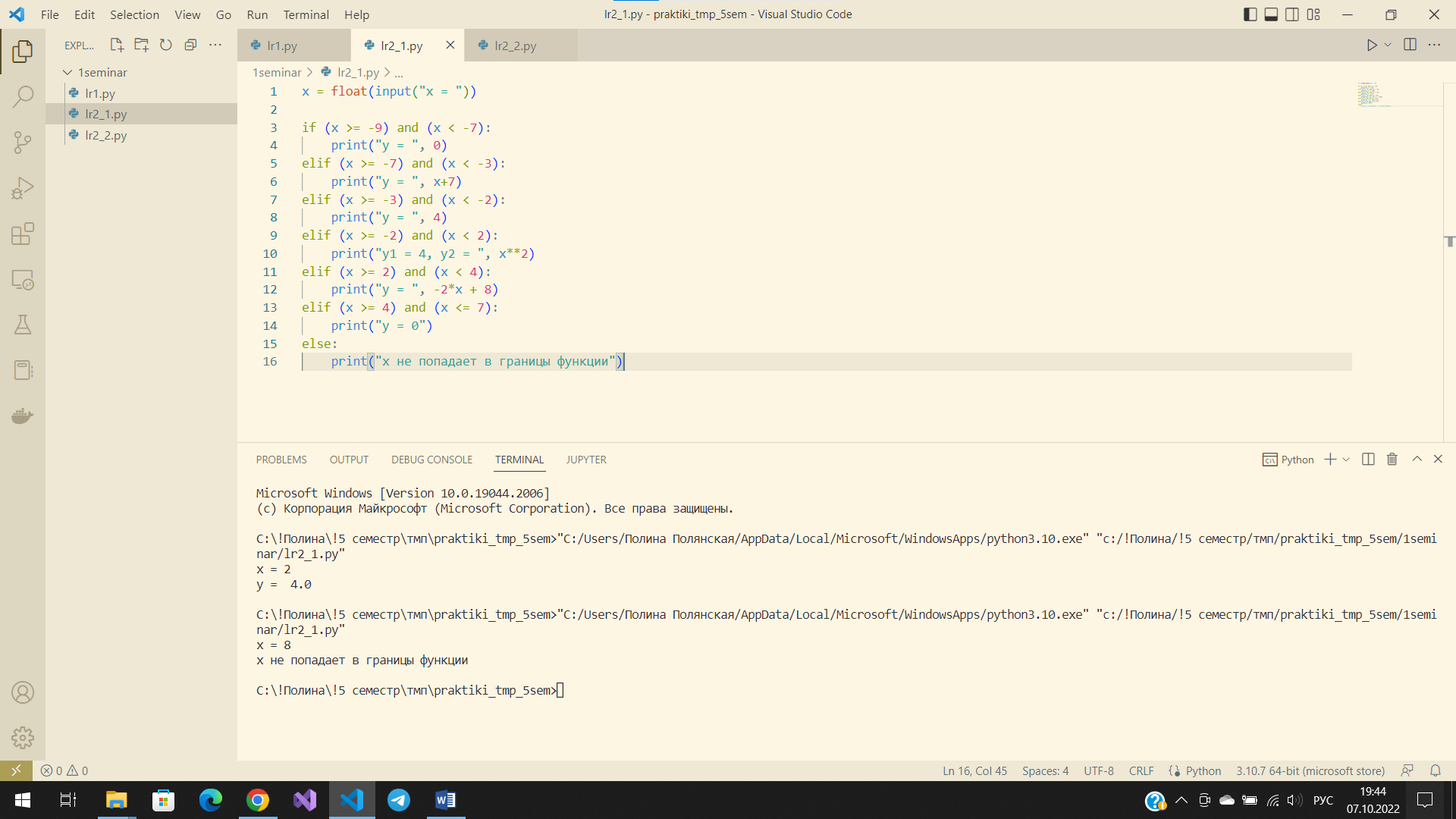
(x\*x + y\*y <= r\*r) and (y <= (x-1)\*\*2) and (x>=1) and (y >=0):

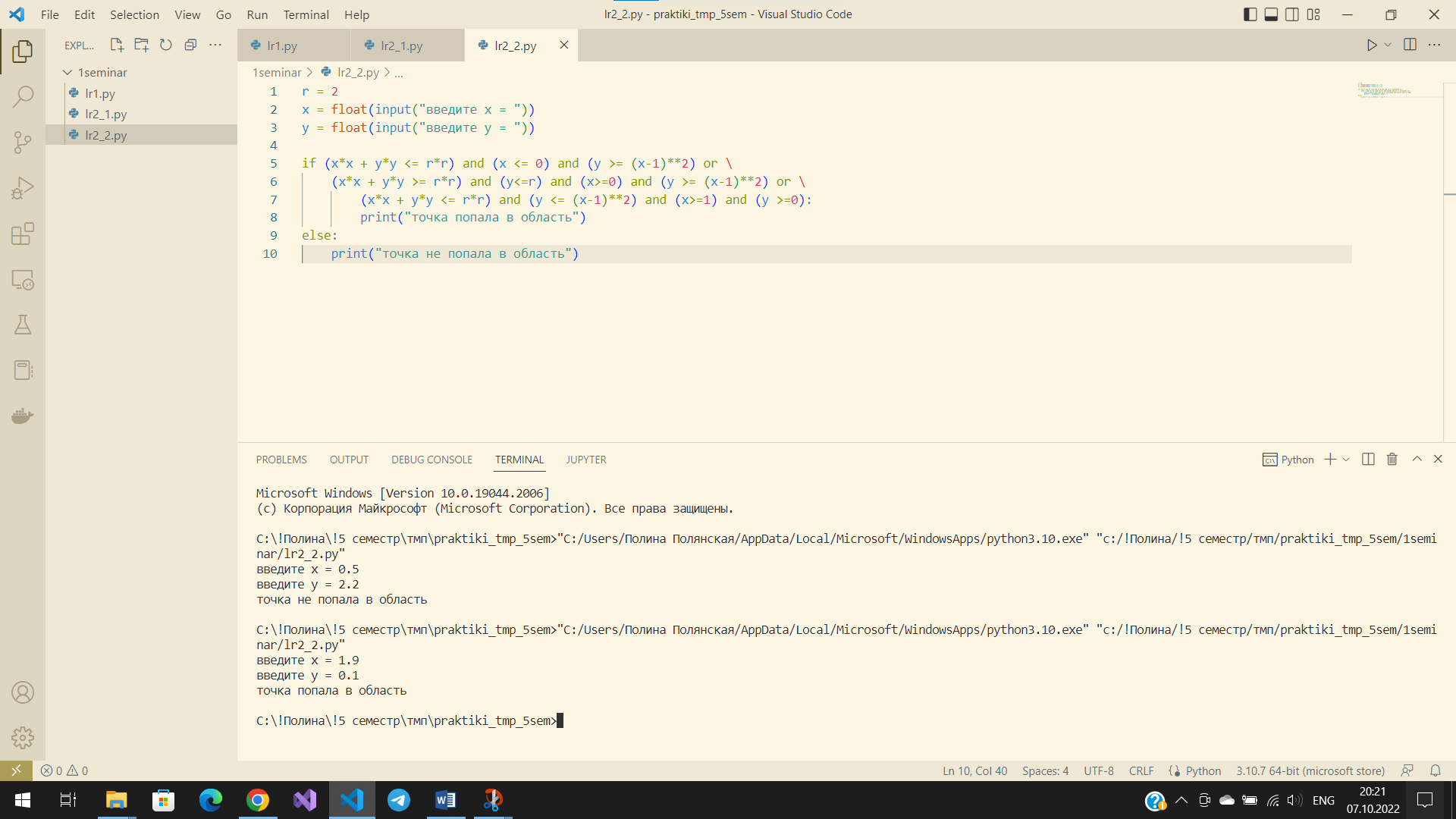
print("точка попала в область")

else:

print("точка не попала в область")

Скриншоты работы программы





**Лабораторная работа №3**

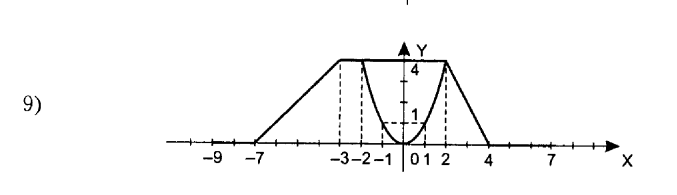
Общее задание

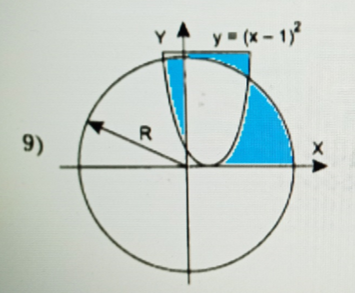
1. Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (см. задание 1 лабораторной работы № 2), на интервале от хнач до хкон с шагом dx. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.
2. Для десяти выстрелов, координаты которых задаются с клавиатуры, вывести текстовые сообщения о попадании в мишень из задания 2 лабораторной работы № 2.
3. Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от хнач до хкон с шагом dх с точностью е. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.

Номер варианта, текст варианта

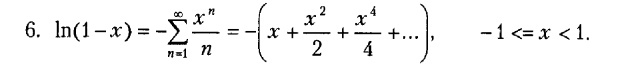
**Номер варианта 9 (для №1№2)**

Текст варианта:





**Номер варианта 6 (для №3)**

****

Описание логической структуры программы

1)

x = float(input("Введите x0 = "))

xk = float(input("Введите xKon = "))

shag = float(input("Введите шаг = "))

s='-----------------------'

print(' Таблица')

print(s)

print('| x | y |')

print(s)

while x<xk:

if (x >= -9) and (x < -7):

y = 0

print('|',"{:5.2f}".format(x),'|', "{:11.4f}".format(y),'|')

print(s)

elif (x >= -7) and (x < -3):

y = x+7

print('|',"{:5.2f}".format(x),'|', "{:11.4f}".format(y),'|')

print(s)

elif (x >= -3) and (x < -2):

y = 4

print('|',"{:5.2f}".format(x),'|', "{:11.4f}".format(y),'|')

print(s)

elif (x >= -2) and (x < 2):

y = 4

y2 = x\*\*2

print('|',"{:5.2f}".format(x),'|', "{:4.2f}".format(y),',', "{:4.2f}".format(y2),'|')

print(s)

elif (x >= 2) and (x < 4):

y = -2\*x + 8

print('|',"{:5.2f}".format(x),'|', "{:11.4f}".format(y),'|')

print(s)

elif (x >= 4) and (x <= 7):

y = 0

print('|',"{:5.2f}".format(x),'|', "{:11.4f}".format(y),'|')

print(s)

else:

print('|',"{:5.2f}".format(x),'|', 'не попадает |')

print(s)

x = x + shag

2)

r = 2

n = int(input("сколько выстрелов будет? = "))

for i in range(10):

x = float(input("введите x = "))

y = float(input("введите y = "))

if (x\*x + y\*y <= r\*r) and (x <= 0) and (y >= (x-1)\*\*2) or \

(x\*x + y\*y >= r\*r) and (y<=r) and (x>=0) and (y >= (x-1)\*\*2) or \

(x\*x + y\*y <= r\*r) and (y <= (x-1)\*\*2) and (x>=1) and (y >=0):

print("при x=",x, " и y=",y,"вы попали в мишень")

else:

print("при x=",x, " и y=",y,"вы НЕ попали в мишень")

3)

#6 вариант

tire = "------------------------"

dx = float(input("введите шаг dx = "))

e = 0.0001

print("точность =",e)

xn = -1

xk = 1

x = xn

print(" таблица ")

print(" x | y | кол-во")

print(tire)

while x < xk:

a = x

s = a

i = 2

while abs(a) > e:

a = (x\*\*i)/i

s = s + a #сумма

i = i + 1 #количество

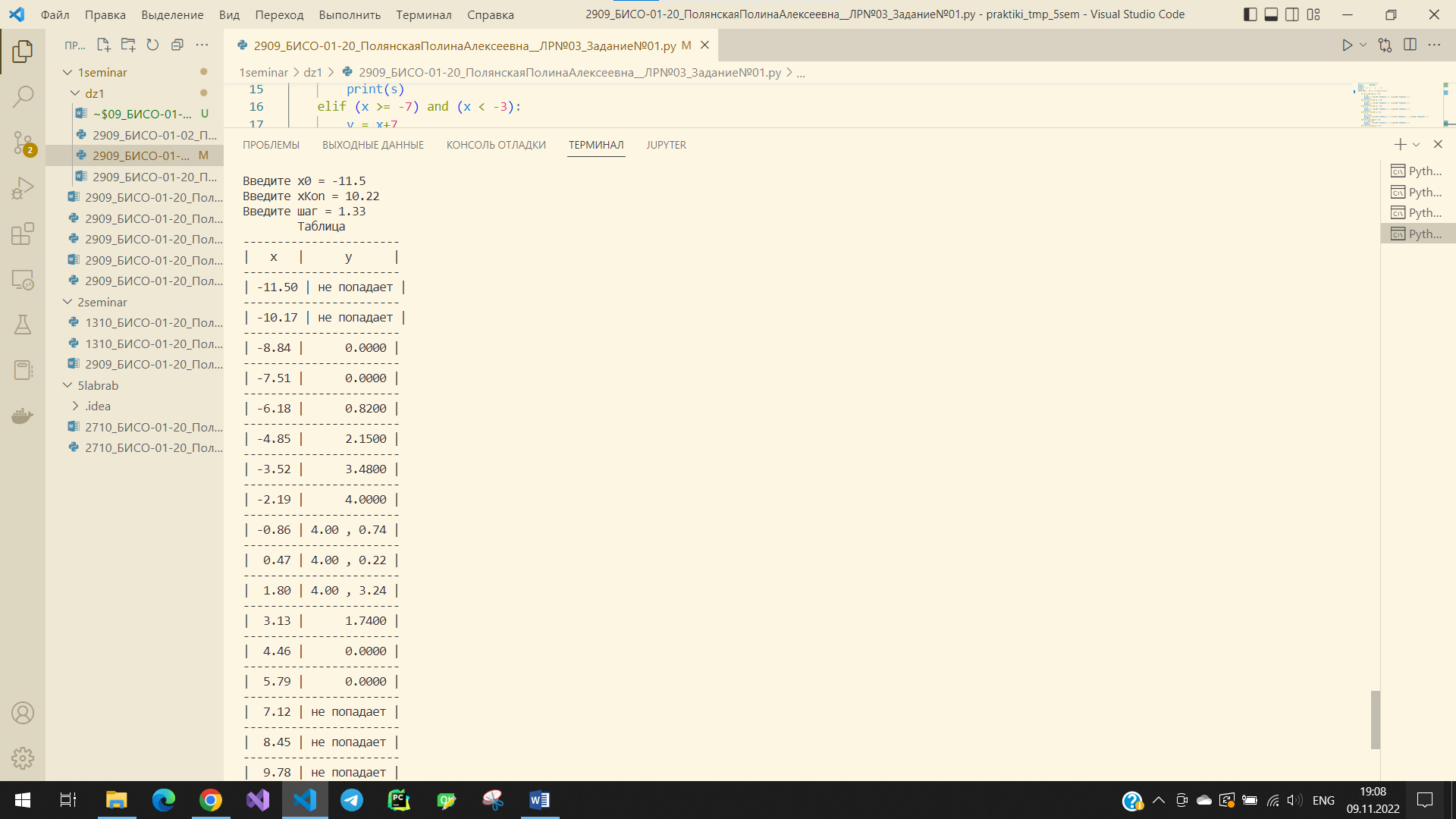
s = -1 \* s

print("{:4.3f}".format(x),"|","{:4.3f}".format(s),"| ",i)

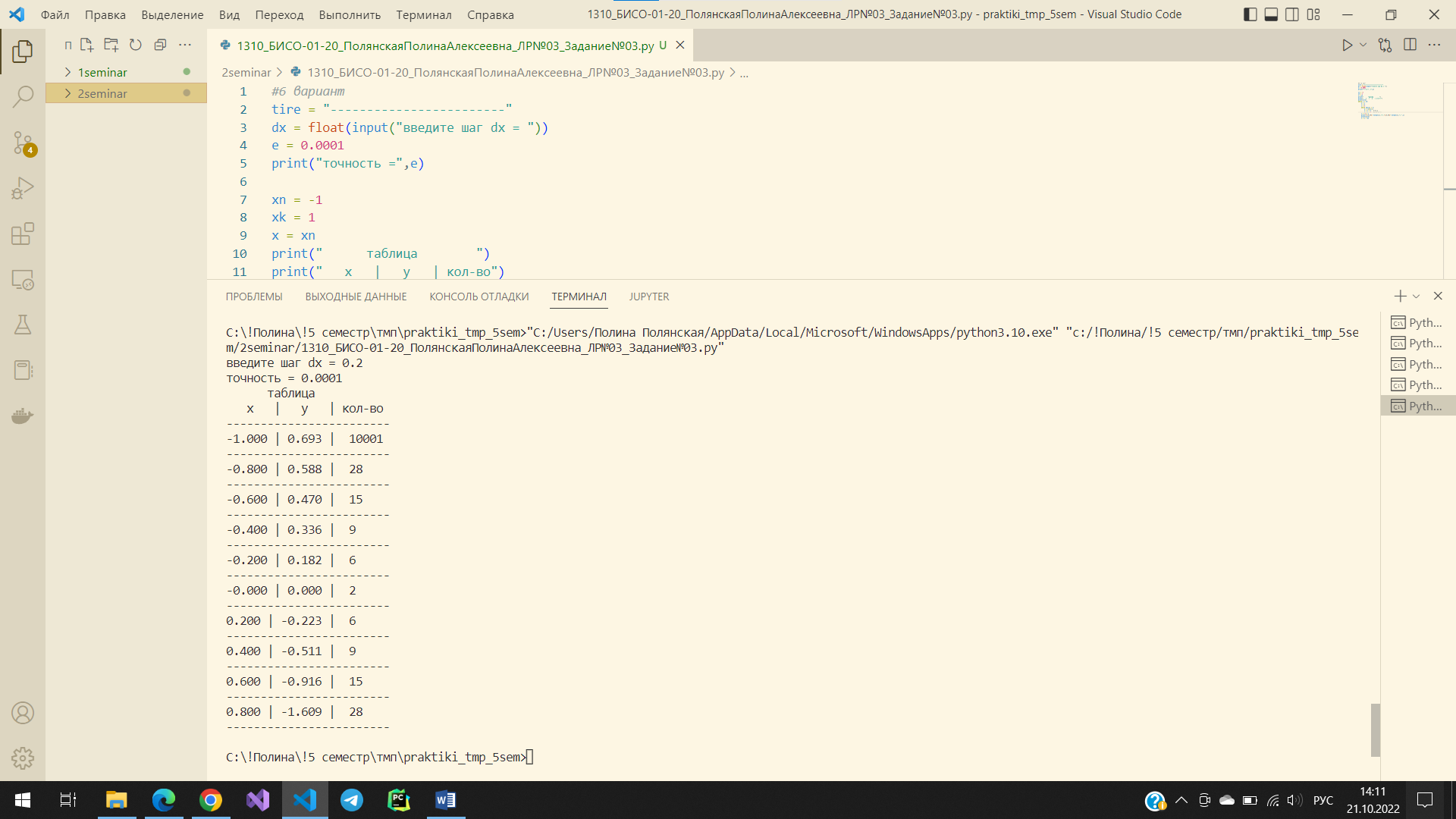
print(tire)

x = x + dx

Скриншоты работы программы

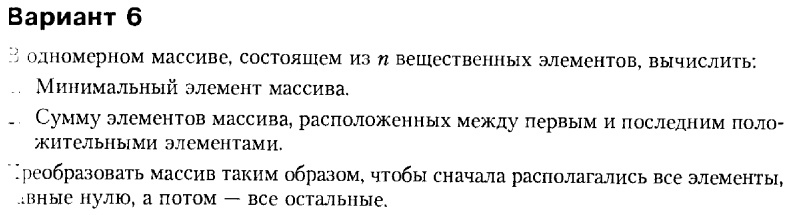






**Лабораторная работа №4**

Общее задание, номер варианта, текст варианта



Выполнить сортировку массива

4)сортировка слиянием

Описание логической структуры программы

# 6 вариант; 4 - сортировка слиянием

import random

def merge\_sort(spisok):

if len(spisok) > 1:

mid = len(spisok)//2

left = spisok[:mid]

right = spisok[mid:]

merge\_sort(left)

merge\_sort(right)

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

spisok[k] = left[i]

i+=1

else:

spisok[k] = right[j]

j+=1

k+=1

while i < len(left):

spisok[k] = left[i]

i+=1

k+=1

while j < len(right):

spisok[k] = right[j]

j+=1

k+=1

n = int(input("введите количество элементов массива: "))

f = int(input("введите 0, если хотите наполнить массив рандомными числами, иначе введите 1: "))

m = list()

for i in range(n):

if f:

a = float(input("введите значение: "))

m.append(a)

else:

a = random.uniform(-100,100)

m.append(a)

print(m)

fi = -1

li = -1

min = m[0]

for i in range(n):

if (fi == -1) and (m[i] > 0):

fi = i

if (m[i]>0):

li = i

if m[i]<min:

min = m[i]

s = 0

for i in range(fi+1, li):

s = s + m[i]

print("сумма эл-тов между первым и последним положительным = ", s)

print("минимальный элемент = ", min)

shi = 0

for i in range(n):

if m[i] == 0:

m[shi], m[i] = m[i], m[shi]

shi = shi + 1

print("сначала 0(если есть), потом остальное: ")

print(m)

#сортировка слиянием

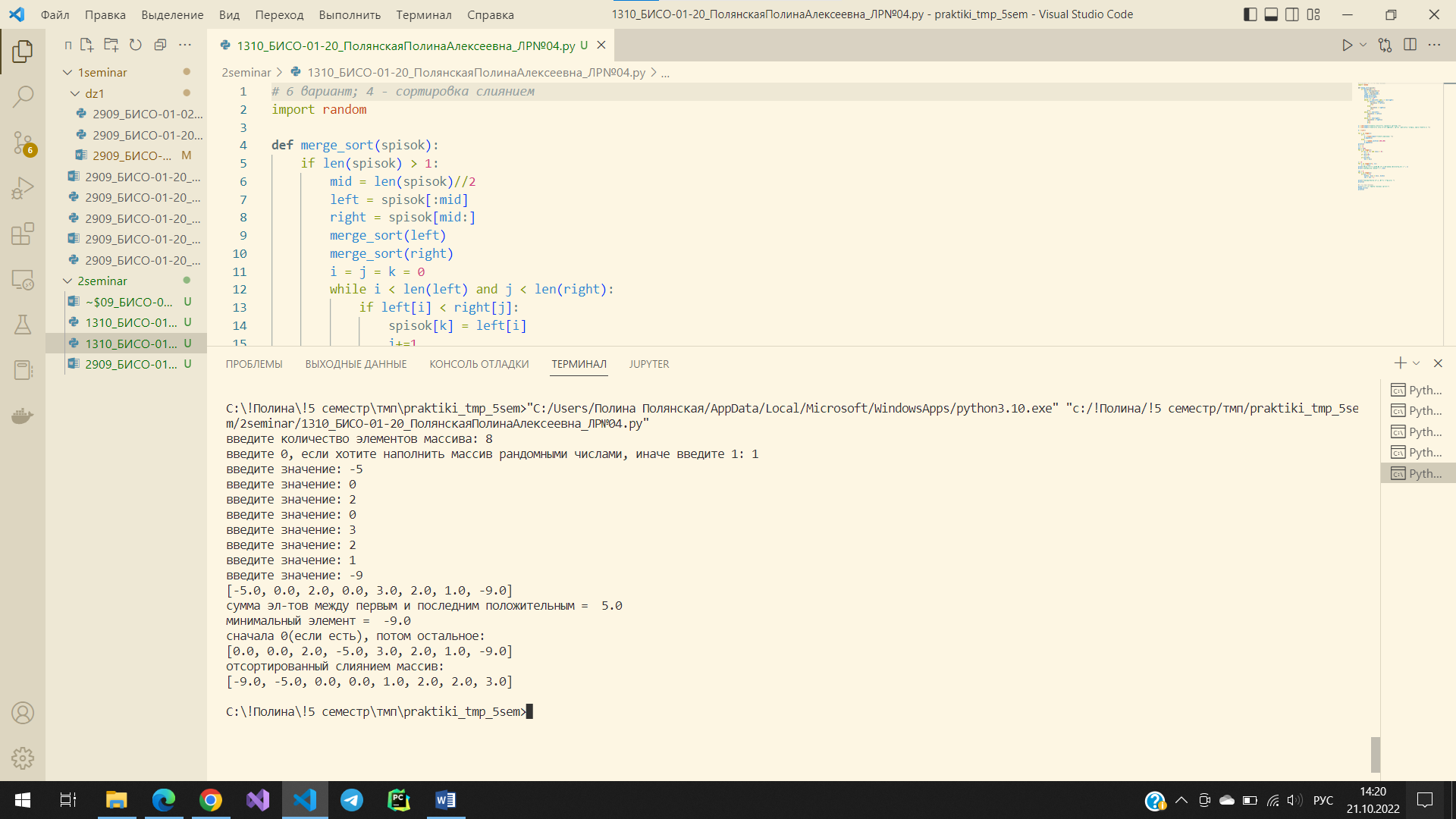
print("отсортированный слиянием массив:")

merge\_sort(m)

print(m)

Скриншоты работы программы





**Лабораторная работа №5**

Общее задание

Размерности массивов следует задать именованными константами. Все необходимые данные должны передаваться подпрограммам в качестве параметров; все величины, используемые только внутри подпрограмм, должны быть описаны как локальные. Использование глобальных переменных в подпрограммах не допускается. Вывод результатов работы подпрограмм должен выполняться в главной программе.

Номер варианта, текст варианта

Вариант 9

Соседями элемента Aij в матрице назовем элементы Аkl, где і - 1 <= k <= i + 1,

j - 1 <= l <= j + 1, (k, l) != (i, j). Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером 10 × 10 (оформить в виде процедуры).

В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали (оформить в виде функции).

Описание логической структуры программы

from random import uniform

n = int(input("сколько строк в матрице? "))

m = int(input("сколько столбцов в матрице? "))

A = []

for i in range(n):

A.append( [0]\*m )

S = []

for i in range(n):

S.append( [0]\*m )

f = int(input("Введите, если: массив наполнить рандомно 0, с клавиатуры 1 "))

if f:

for i in range(n):

for j in range(m):

A[i][j] = float(input(f'введите элемент:A[{i}][{j}]='))

else:

for i in range(n):

for j in range(m):

A[i][j] = uniform(1,30)

def sglazhivanie(A,S,n,m):

for i in range(n):

for j in range(m):

s = 0;

kolv = 0

for k in range(i-1,i+2):

for l in range(j-1,j+2):

if ((k,l) != (i,j)) and (k>=0) and (l>=0) and (k<n) and (l<m):

s = s + A[k][l]

kolv = kolv + 1

if kolv>0:

s = s / kolv

S[i][j] = s

sglazhivanie(A,S,n,m)

def sum\_abs\_niz\_gd(S, n, m):

summ = 0

for i in range(n):

for j in range(m):

if i>j:

summ = summ + abs(S[i][j])

return summ

print("A:")

for i in range(n):

for j in range(m):

print(A[i][j], end=" ")

print()

print("S:")

for i in range(n):

for j in range(m):

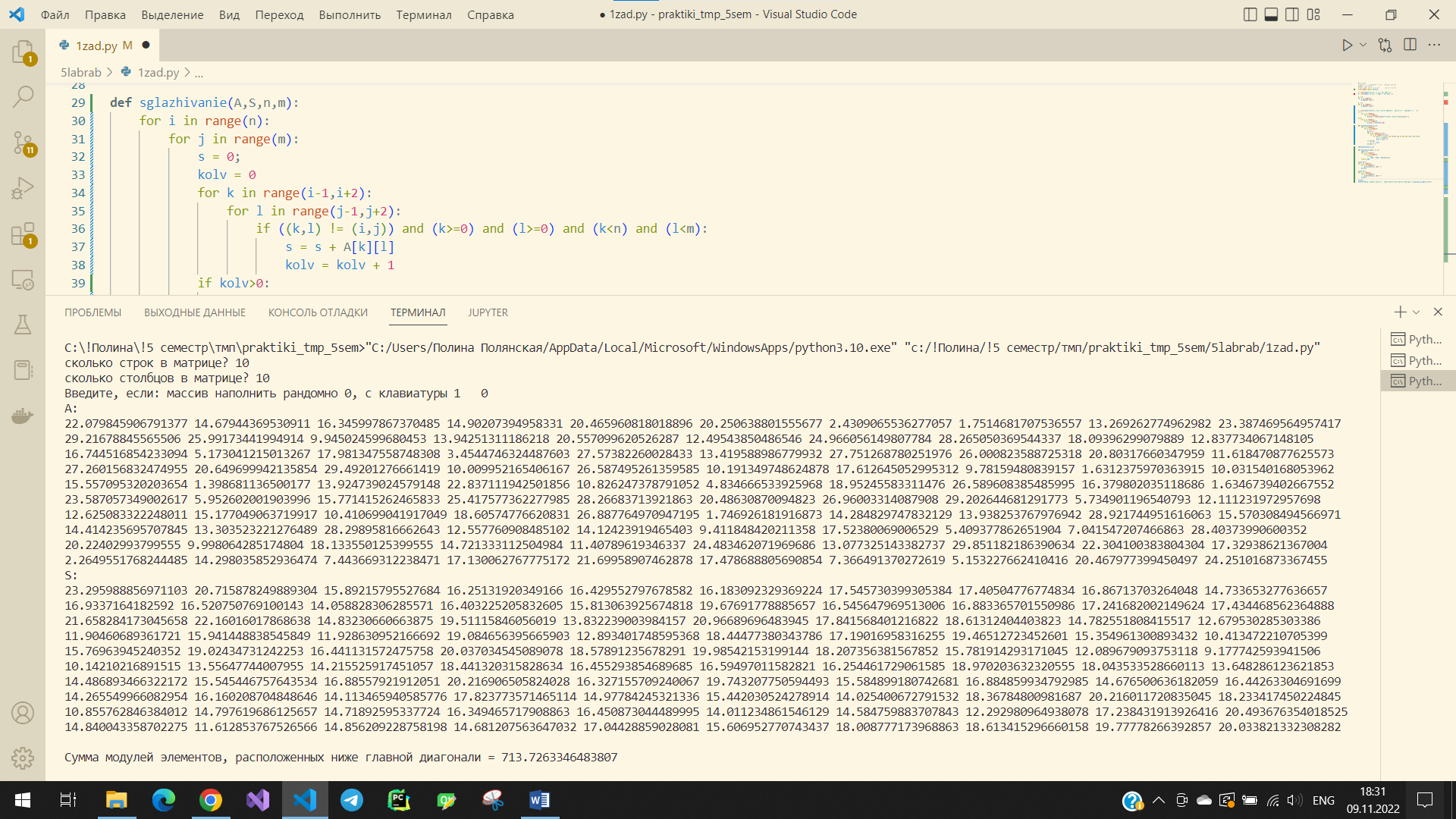
print(S[i][j], end=" ")

print()

print()

print(f"Сумма модулей элементов, расположенных ниже главной диагонали = {sum\_abs\_niz\_gd(S,n,m)}")

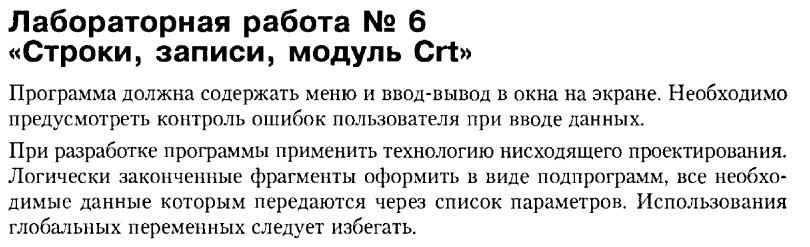
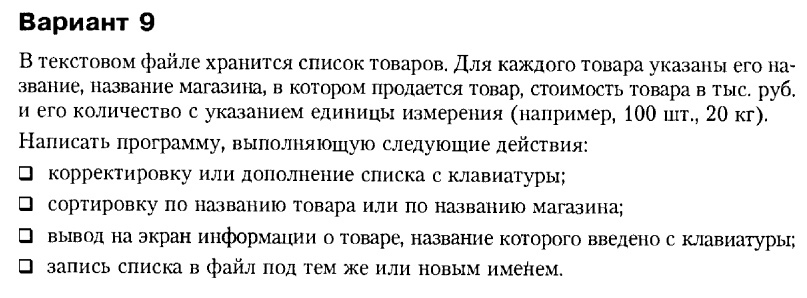
Скриншоты работы программы



**Лабораторная работа №6**

Общее задание, номер варианта, текст варианта

9 вариант.

Описание логической структуры программы

**class Tovar:**

**def \_\_init\_\_(self, \_name, \_shop, \_price, \_kolvo):**

**self.name = \_name**

**self.shop = \_shop**

**self.price = \_price**

**self.kolvo = \_kolvo**

**self.oshib = 0**

**def getName(self):**

**return self.name**

**def getShop(self):**

**return self.shop**

**def getPrice(self):**

**return self.price**

**def getKolvo(self):**

**return self.kolvo**

**def setName(self,\_name):**

**self.name = \_name**

**def setShop(self,\_shop):**

**self.shop = \_shop**

**def setPrice(self,\_price):**

**self.price = \_price**

**def setKolvo(self,\_kolvo):**

**self.kolvo = \_kolvo**

**class Menu:**

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

**self.oshib = 0**

**self.list = []**

**f = True**

**while f:**

**print("Новый файл - 1")**

**print("Существующий файл - 2")**

**try:**

**m = int(input("Введите цифру "))**

**if m == 1 or m == 2:**

**f = False**

**except ValueError:**

**continue**

**if m == 2:**

**import os**

**for root, dirs, files in os.walk("."):**

**for fnm in files:**

**if fnm.endswith(".txt"):**

**print(fnm)**

**fa = True**

**while fa:**

**file\_name = input("введите название файла: ")**

**try:**

**with open(file\_name, 'r') as file:**

**if file != None:**

**try:**

**arr = file.read().split('\n')**

**for i in arr:**

**temp = i.split(',')**

**name = temp[0]**

**shop = temp[1]**

**price = temp[2]**

**kolvo = temp[3]**

**self.list.append(Tovar(name,shop,price,kolvo))**

**fa = False**

**self.oshib = 0**

**except:**

**print("Найдена ошибка при обработке файла")**

**else:**

**print("Найдена ошибка при открытии файла")**

**#return**

**except:**

**print("найдена ошибка при открытии файла")**

**self.oshib = 1**

**#return**

**def save(self):**

**if not len(self.list):**

**print("Нет данных ")**

**return**

**listvr = []**

**for i in self.list:**

**name = i.getName()**

**shop = i.getShop()**

**price = i.getPrice()**

**kolvo = i.getKolvo()**

**tempString = ','.join([name,shop,price,kolvo])**

**listvr.append(tempString)**

**print()**

**import os**

**for root, dirs, files in os.walk("."):**

**for fnm in files:**

**if fnm.endswith(".txt"):**

**print(fnm)**

**print()**

**file\_name = input("введите название файла (сохранить как, можете ввести новое): ")**

**with open(file\_name, 'w') as file:**

**string = '\n'.join(listvr)**

**file.write(string)**

**def navigation(self):**

**while True:**

**if self.oshib:**

**return**

**print('Информация о товаре - 1')**

**print('Добавить товар - 2')**

**print('Изменить информацию о товаре - 3')**

**print('Вывести товары в отсортированном виде по названию товара - 4')**

**print('Вывести товары в отсортированном виде по названию магазина - 5')**

**print('Сохранить файл как - 6')**

**print('Выйти из программы - 0')**

**m = int(input('Введите номер пункта: '))**

**if (m >= 0 and m < 7):**

**if m == 1:**

**try:**

**#self.info\_tov()**

**self.info\_tov(tov=input("Введите название товара: "))**

**except:**

**print("Такого товара нет ")**

**if m == 2:**

**self.dobavit()**

**if m == 3:**

**self.izmenit()**

**if m == 4:**

**self.sortirovka\_tov()**

**if m == 5:**

**self.sortirovka\_magaz()**

**if m == 6:**

**self.save()**

**if m==0:**

**return**

**else:**

**print("неверный номер")**

**def info\_tov(self,\*\*kwargs):**

**if not len(self.list):**

**print("Товаров в данном файле нет")**

**return**

**if 'tov' in kwargs.keys():**

**f = 0**

**prod\_List = []**

**for kortej in self.list:**

**if kortej.getName() == kwargs['tov']:**

**print("Название товара - ", kortej.getName())**

**print("Магазин - ", kortej.getShop())**

**print("Цена - ", kortej.getPrice())**

**print("Количество - ", kortej.getKolvo())**

**print()**

**f = 1**

**prod\_List.append(kortej)**

**if f == 0:**

**print("Такого товара не существует ")**

**return prod\_List**

**else:**

**for kortej in self.list:**

**print("Название товара - ", kortej.getName())**

**print("Магазин - ", kortej.getShop())**

**print("Цена - ", kortej.getPrice())**

**print("Количество - ", kortej.getKolvo())**

**print()**

**return**

**def dobavit(self):**

**name = input("Введите название товара: ")**

**shop = input("Введите название магазина: ")**

**while True:**

**price = input("Цена: ")**

**try:**

**priceval = int(price)**

**price = str(priceval)**

**break**

**except ValueError:**

**print("Значение цены должно быть числовым")**

**while True:**

**kolvo = input("Количество: ")**

**try:**

**kolvoVal = int(kolvo)**

**while True:**

**zn = input("Укажите количество (шт., г., кг., м., л.):")**

**setzn = {'кг.', 'м.', 'г.', 'л.', 'шт.'}**

**if zn in setzn:**

**break**

**else:**

**print("Неверная единица измерения ")**

**except ValueError:**

**print("Значение количества должно быть числовым")**

**kolvo = str(kolvoVal) + ' ' + zn**

**break**

**self.list.append(Tovar(name, shop, price, kolvo))**

**self.save()**

**def izmenit(self):**

**self.info\_tov()**

**name = input("Название товара: ")**

**if name == '':**

**print("Вы ничего не ввели")**

**return**

**tov\_List = self.info\_tov(tov=name)**

**if tov\_List == None:**

**return**

**m = 0**

**if len(tov\_List) > 1:**

**while True:**

**try:**

**m = int(input("Введите номер товара:"))**

**break**

**except ValueError:**

**continue**

**try:**

**print()**

**vybtov = tov\_List[m-1]**

**print("Название: ", vybtov.getName())**

**print("Магазин: ", vybtov.getShop())**

**print("Цена: ", vybtov.getPrice())**

**print("Кол-во: ", vybtov.getKolvo())**

**print()**

**except:**

**print("Вы неправильно ввели данные ")**

**self.izmenit()**

**print('Изменить название товара - 1')**

**print('Изменить название магазина - 2')**

**print('Изменить цену товара - 3')**

**print('Изменить количество товара - 4')**

**while True:**

**try:**

**m = int(input('Введите номер варианта: '))**

**if m in [1,2,3,4]:**

**break**

**except ValueError:**

**continue**

**if m == 1:**

**name = input("Название товара: ")**

**vybtov.setName(name)**

**if m == 2:**

**shop = input("Название магазина: ")**

**vybtov.setShop(shop)**

**if m == 3:**

**while True:**

**price = input("Цена: ")**

**try:**

**priceval = int(price)**

**price = str(priceval)**

**break**

**except ValueError:**

**print("Значение цены должно быть числовым")**

**vybtov.setPrice(price)**

**if m == 4:**

**while True:**

**kolvo = input("Количество: ")**

**try:**

**kolvoVal = int(kolvo)**

**while True:**

**zn = input("Укажите количество (шт., г., кг., м., л.):")**

**setzn = {'кг.', 'м.', 'г.', 'л.', 'шт.'}**

**if zn in setzn:**

**break**

**else:**

**print("Неверная единица измерения ")**

**kolvo = str(kolvoVal) + ' ' + zn**

**break**

**except ValueError:**

**print("Значение количества должно быть числовым ")**

**vybtov.setKolvo(kolvo)**

**self.save()**

**def sortirovka\_magaz(self):**

**self.list = sorted(self.list,key=lambda x:x.getShop())**

**self.info\_tov()**

**def sortirovka\_tov(self):**

**self.list = sorted(self.list,key=lambda x:x.getName())**

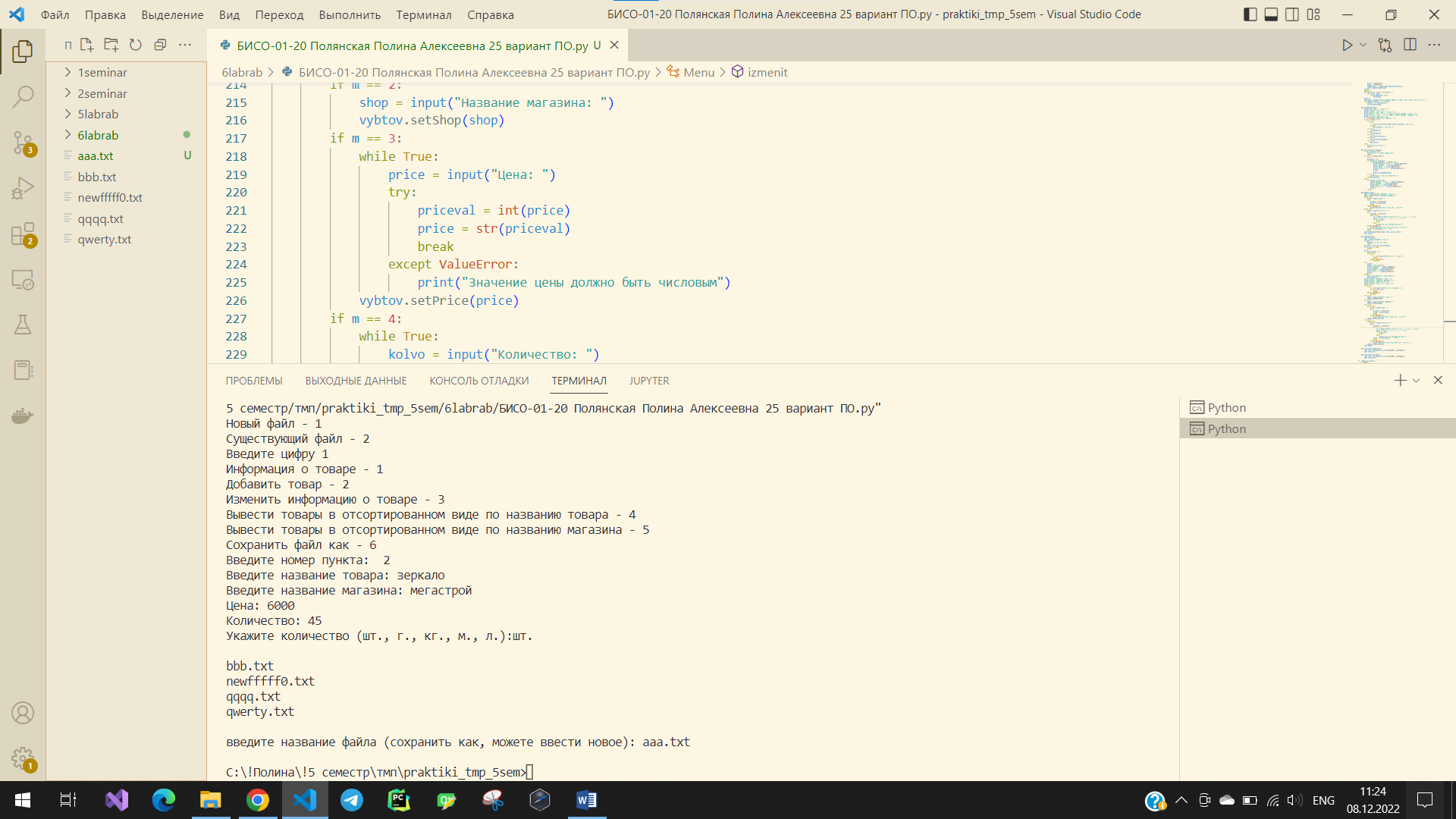
**self.info\_tov()**

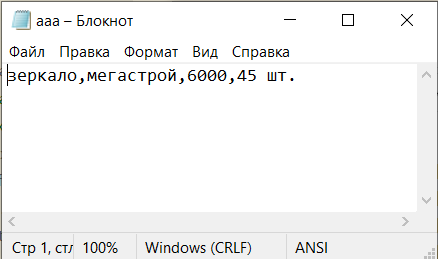
**if \_\_name\_\_=="\_\_main\_\_":**

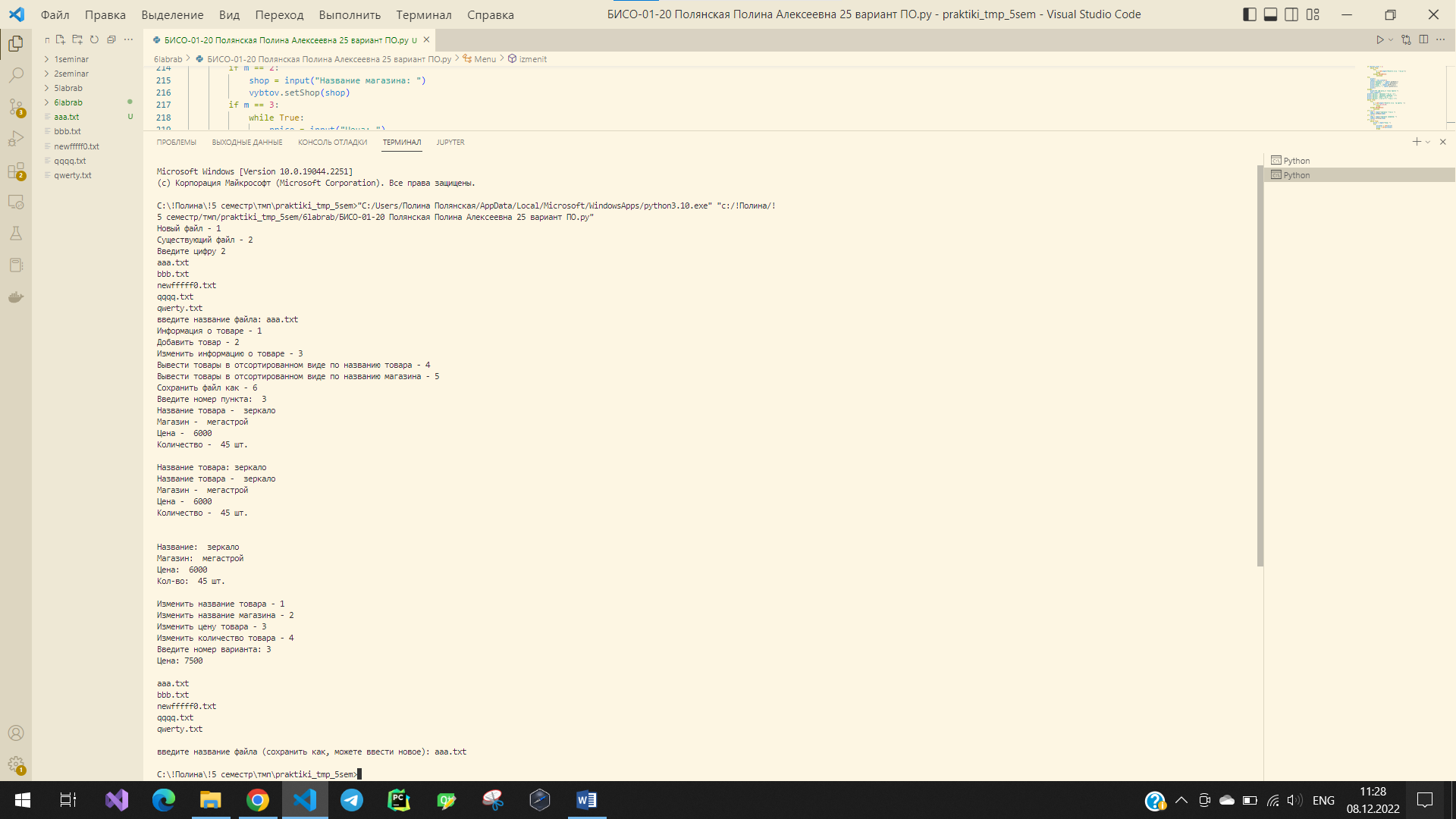
**A = Menu()**

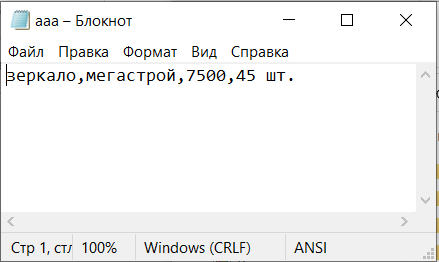
**A.navigation()**

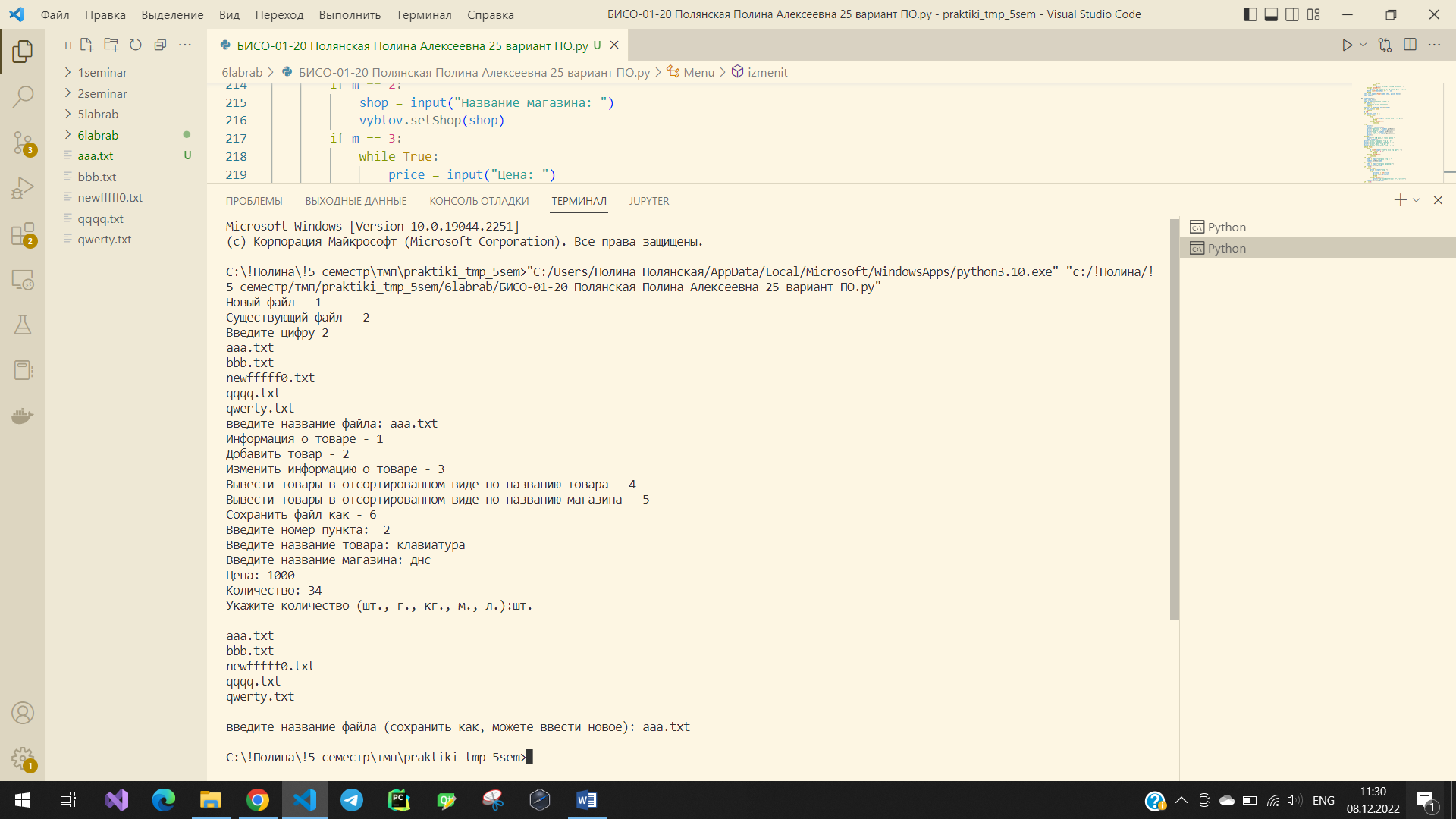
Скриншоты работы программы

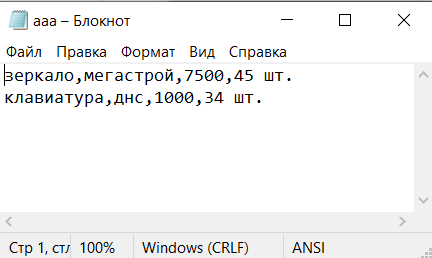


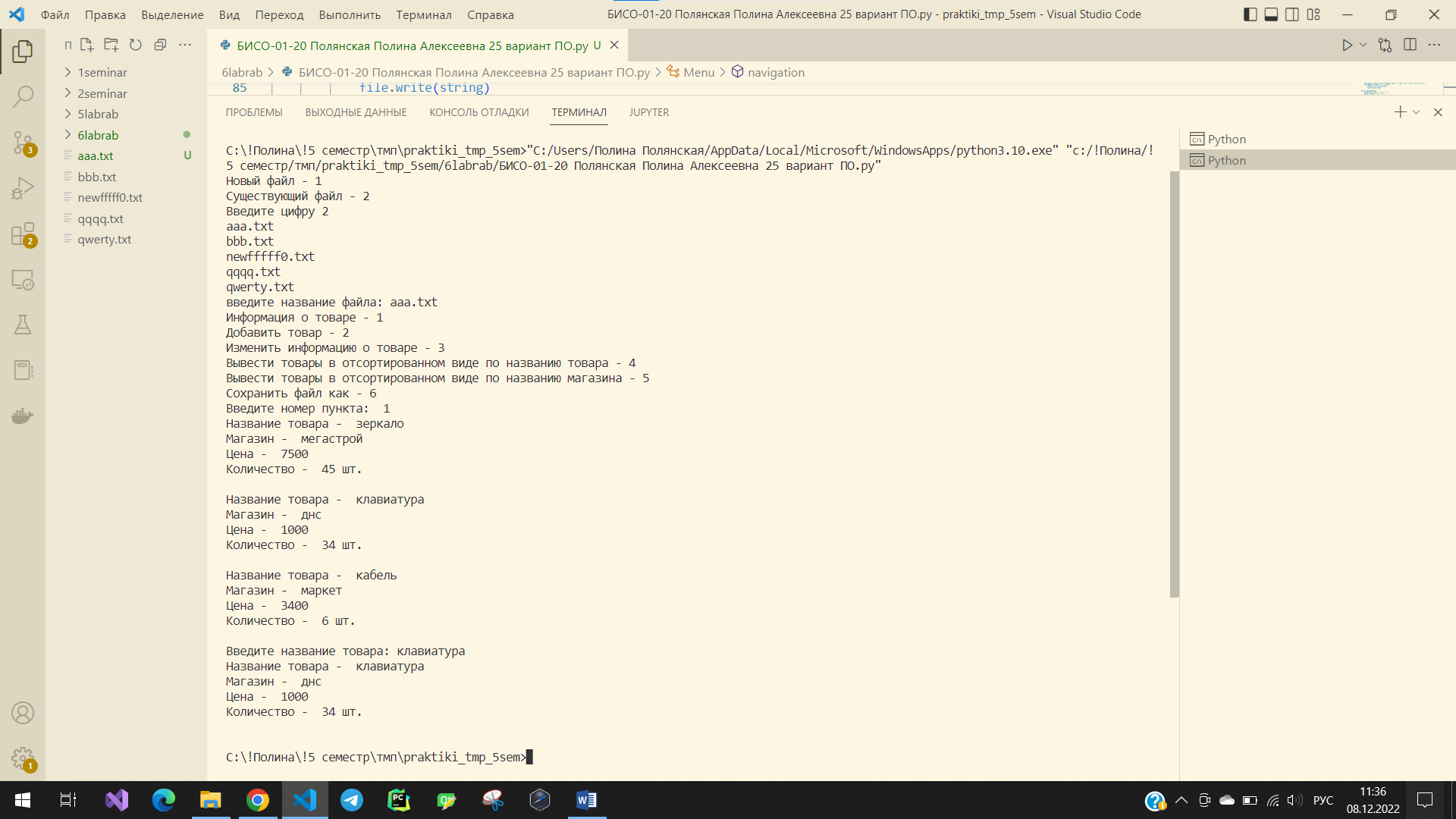


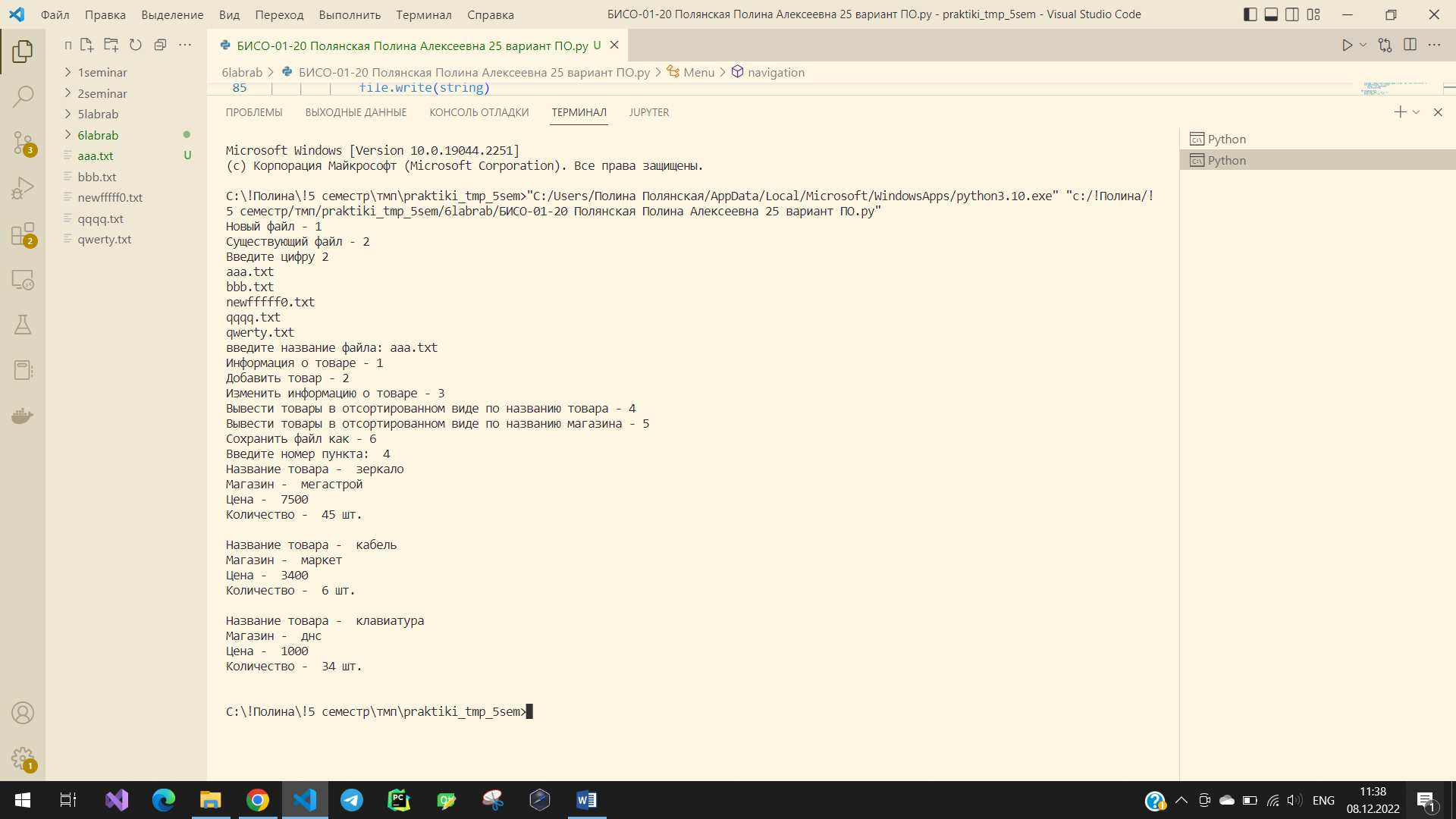


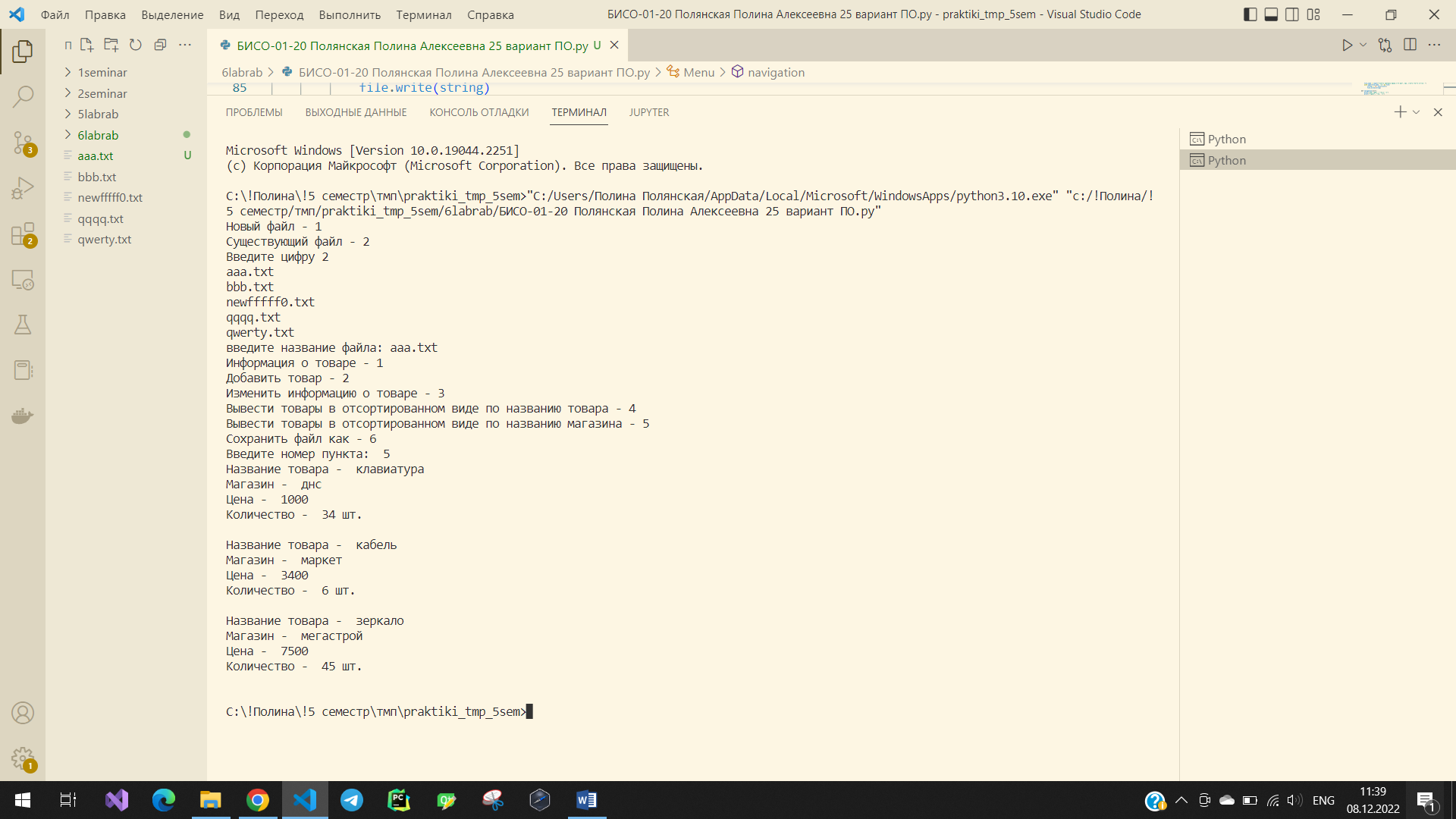


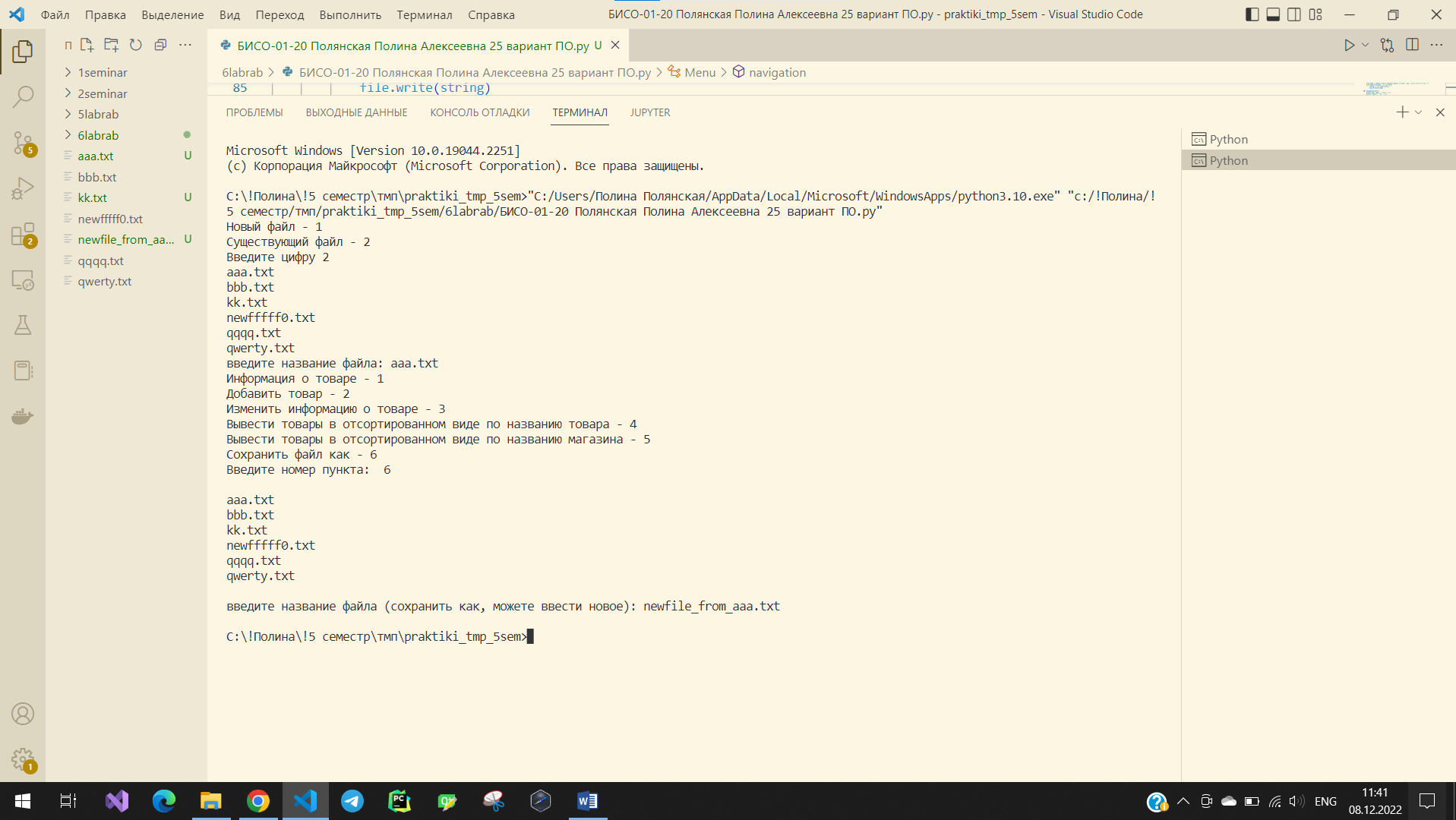


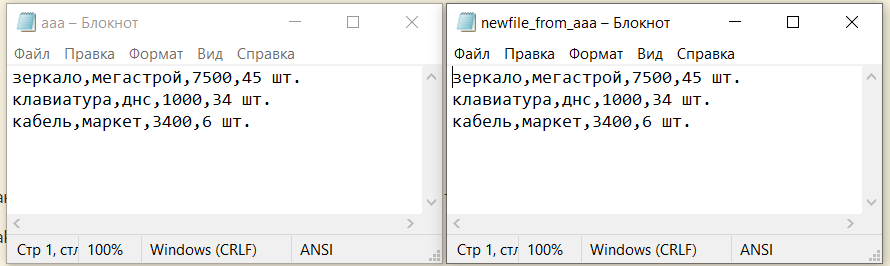


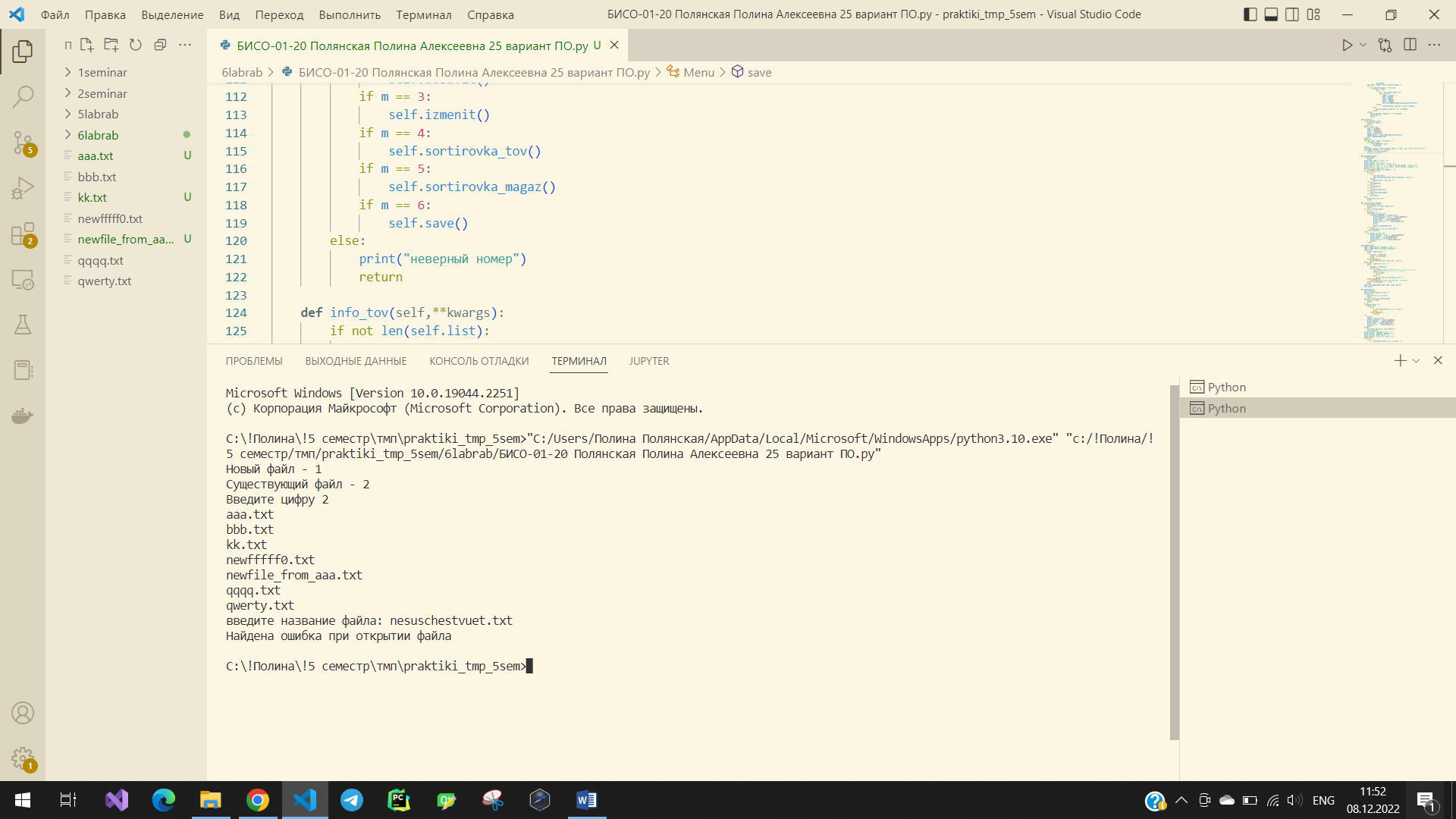








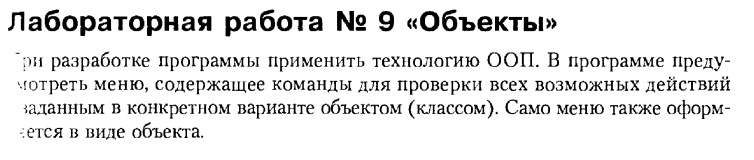


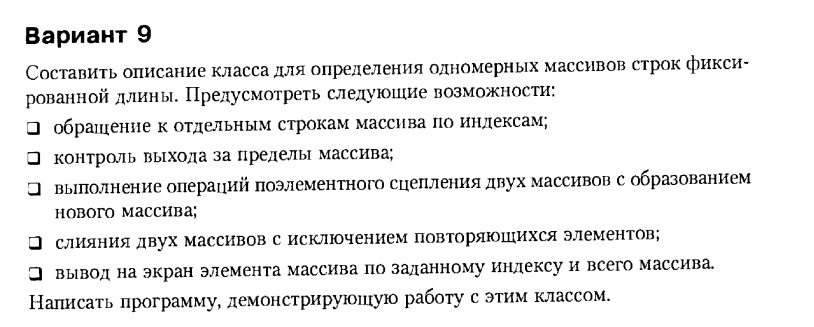


**Лабораторная работа №9**

Общее задание, номер варианта, текст варианта

9 вариант.





**Листинг программы**

class Menu:

    def \_\_init\_\_(self):

        self.\_\_vibor = 0

        self.AA = []

        while True:

            print("задать массив строк - 1")

            print("изменить массив - 2")

            print("поэлементная конкатенация массивов - 3")

            print("слияние массивов - 4")

            print("вывести элемент массива на печать - 5")

            print("вывести массивы на печать - 6")

            print("выйти из программы - 0")

            self.\_\_vibor = int(input(">> "))

            if self.\_\_vibor == 0:

                break

            elif self.\_\_vibor == 1:

                length = int(input("количество строк в массиве: "))

                self.A = Yozh(length)

                i=0

                for i in range(length):

*#str\_length = int(input(""))*

                    q = input("введите строку (прекратить - !)  ")

                    if q=='!':

                        break

                    else:

                        self.A.\_\_setitem\_\_(i, q)

                        i=i+1

                self.AA.append(self.A)

            elif self.\_\_vibor == 2:

                indM = int(input("введите индекс МАССИВА "))

                if indM > len(self.AA):

                    print("неверный индекс массива")

                    continue

                indM = indM - 1

                ind = int(input("введите индекс элемента"))

                ind = ind - 1

                zna = input("введите строку ")

                self.AA[indM].\_\_setitem\_\_(ind, zna)

            elif self.\_\_vibor == 3:

                indM1 = int(input("введите индекс одного МАССИВА "))

                if indM1 > len(self.AA):

                    print("неверный индекс массива")

                    continue

                indM1 = indM1 - 1

                indM2 = int(input("введите индекс другого МАССИВА "))

                if indM2 > len(self.AA):

                    print("неверный индекс массива")

                    continue

                indM2 = indM2 - 1

                C = self.AA[indM1].poel\_concat(self.AA[indM2])

                self.AA.append(C)

            elif self.\_\_vibor == 4:

                indM1 = int(input("введите индекс одного МАССИВА "))

                if indM1 > len(self.AA):

                    print("неверный индекс массива")

                    continue

                indM1 = indM1 - 1

                indM2 = int(input("введите индекс другого МАССИВА "))

                if indM2 > len(self.AA):

                    print("неверный индекс массива")

                    continue

                indM2 = indM2 - 1

                C = self.AA[indM1].sliyan(self.AA[indM2])

                self.AA.append(C)

            elif self.\_\_vibor == 5:

                indM = int(input("введите индекс МАССИВА "))

*#indM = indM - 1*

                ind = int(input("введите индекс элемента"))

                ind = ind - 1

*#print(len(self.AA))*

                if indM <= len(self.AA):

                    indM = indM - 1

                    self.AA[indM].print(ind)

                else:

                    print("неверный индекс массива")

            elif self.\_\_vibor == 6:

                for B in self.AA:

                    B.print\_array()

            else:

                print("вы не то ввели")

            print()

class YozhException(Exception):

    pass

class Yozh:

    """описание класса для определения одномерных массивов строк фиксированной длины"""

    def \_\_init\_\_(self, length: int, str\_length=35):

*#length: длина массива*

*#str\_length: длина строки в массиве*

        if not isinstance(length, int):   *#явл ли объектом класса*

            raise YozhException()

        if not isinstance(str\_length, int):   *#явл ли объектом класса*

            raise YozhException()

        self.\_\_length = length

        self.\_\_str\_length = str\_length

        self.\_\_yozhiki = ['' for \_ in range(length)]

    def \_\_setitem\_\_(self, index: int, value: str):

        if not isinstance(index, int):

            raise YozhException('int != {}'.format(type(index)))

        if not isinstance(value, str):

            raise YozhException('')

        if len(value) > self.\_\_str\_length:

            raise YozhException('строка слишком длинная')

        if (self.\_\_check\_overflow(index)):

            self.\_\_yozhiki[index] = value

    def \_\_getitem\_\_(self, index: int):

        if not isinstance(index, int):

            raise YozhException('int != {}'.format(type(index)))

        if (self.\_\_check\_overflow(index)):

            return self.\_\_yozhiki[index]

    def \_\_check\_overflow(self, index: int):

        if not (0 <= index < len(self.\_\_yozhiki)):

*#raise YozhException('выход за границы массива')*

*#pass*

            print("выход за границы массива")

            return 0

        else:

            return 1

    def print(self, index: int):      *#печать переделать*

        """печать (вывод на экран) элементов массива"""

*#print(self[index])*

        print(self.\_\_getitem\_\_(index))

    def print\_array(self):

        """печать (вывод на экран) всего массива"""

        print(self)

    def \_\_str\_\_(self):

        return ' '.join(self.\_\_yozhiki)

    def poel\_concat(self, B: 'Yozh'):

        lenA = self.\_\_length

        lenB = B.\_\_length

        n = min(lenA, lenB)

        C = Yozh(n)

        for i in range(n):

            stro = self.\_\_getitem\_\_(i) + B.\_\_getitem\_\_(i)

            C.\_\_setitem\_\_(i, stro)

        return C

    def sliyan(self, B: 'Yozh'):

        lenA = self.\_\_length

        lenB = B.\_\_length

        setA = set()

        setB = set()

        for i in range(lenA):

            q = self.\_\_getitem\_\_(i)

            setA.add(q)

        for i in range(lenB):

            q =  B.\_\_getitem\_\_(i)

            setB.add(q)

        setC = setA | setB

        n = len(setC)

        C = Yozh(n)

        i = 0

        for q in setC:

            C.\_\_setitem\_\_(i, q)

            i = i + 1

        return C

    def add(self, par):

        print(self.\_\_yozhiki)

        print(self.\_\_length)

        if '' in self.\_\_yozhiki:

            self.\_\_yozhiki.append(par)

        else:

            print(f"{par} не был записан - превышена длина.")

me = Menu()

**Скрины работы программы**

