Зертханалық жұмыс №13

**Тақырыбы:** Модульдермен жұмыс: import және from нұсқауларымен құру, қосу

**Жұмыс мақсаты**: Модульдерді зерттеу және модульдермен жұмыс істеу

***Math кітапханасы (модуль)***

Python стандартты жеткізіліміне math кітапханасы кіреді, оның құрамында жиі қолданылатын математикалық функциялар көп.

Бұл модульмен жұмыс істеу үшін алдымен оны импорттау керек.

**import math**

Math Модулінің жиі қолданылатын функцияларын қарастырыңыз:

|  |  |
| --- | --- |
| math.ceil(x) | X-тен үлкен бүтін санды қайтарады |
| math.fabs(x) | X санының абсолютті мәнін қайтарады |
| math.factorial(x) | Факториалды x есептейді |
| math.floor(x) | X-тен кіші бүтін санды қайтарады |
| math.exp(x) | E\*\*x есептейді |
| math.log2(x) | 2 негіз бойынша логарифм |
| math.log10(x) | 10 негіз бойынша логарифм |
| math.log(x[, base]) | Әдепкі бойынша, ол логарифмді e негізінде есептейді, қосымша Логарифм негізін көрсетуге болады |
| math.pow(x, y) | X мәнін y дәрежесінде есептейді |
| math.sqrt(x) | X-тен квадрат түбір |

Жоғарыда аталған функцияларды сандарға қолдану мысалы

Бағдарламада 4 айнымалы анықталған - a, b, c, d, олардың әрқайсысы бүтін сан немесе нақты немесе теріс.

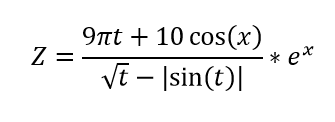
Print () командасы бағдарламаны орындау кезінде әр айнымалының мәнін экранға шығарады.

Math модулі функциясының нәтижесі z айнымалысына орналастырылады.

Содан кейін print()командасы хабарламаны пайдаланылған функция және оның дәлелі және оны орындау нәтижесі ретінде көрсетеді.

Мысал

Берілген айнымалылар үшін берілген арифметикалық өрнекті есептейтін бағдарламаны жазу. Айнымалыларды енгізу пернетақтадан жүзеге асырылады. Нәтижені үтірден кейін 2 таңбамен көрсетіңіз.



x=10; t=1

*Шешім*

Алдымен math модулін импорттаңыз. Ол үшін import math пәрменін пайдаланыңыз.

Содан кейін бүтін X және t типіндегі екі айнымалының мәндерін енгізу керек.

Деректерді енгізу үшін енгізу пәрмені қолданылады, бірақ бүтін сандар берілгендіктен, алдымен айнымалылардың түрін анықтау керек: x=int (), t=int().

Айнымалылардың түрін анықтағаннан кейін оларды енгізу керек, ол үшін Int() командасының жақшаларына input () командасын жазу керек.

X айнымалысы үшін бұл келесідей: x=int (input ("мәнді енгізу кезіндегі хабарлама")).

Айнымалы үшін T ұқсас: t=int (input ("мәнді енгізу кезіндегі хабарлама")).

Келесі қадам-арифметикалық өрнекті құрастыру, оның нәтижесі z айнымалысына орналастырылады.

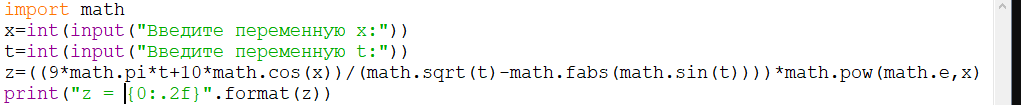
Алдымен алымын жазыңыз. Ол келесідей болады: 9 \* math.pi\*t+10\*math.cos(x).

Содан кейін сіз бөліндіні жазу керек, ал нумератор бөлгішке бөлінетініне назар аударамыз, сондықтан алымы мен бөлгішті жақшаға () салып, олардың арасына бөлу /белгісін жазу керек.

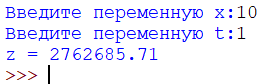
Бұл келесідей болады: (9 \* math.pi\*t+10\*math.cos(x))/(math.sqrt(t)-math.fabs(math.sin(t))).

Соңғы қадам-фракцияны экспонентке x дәрежесіне көбейту.

Бүкіл бөлшек көбейтілгендіктен, құрастырылған өрнек жақшаға салынуы керек (), содан кейін math функциясын жазыңыз.pow(math.e,x).



Нәтижесі:



**Random модулі**

Random модулі кездейсоқ сандардың пайда болуын басқарады. Оның негізгі функциялары:

random(): гкездейсоқ санды 0.0-ден 1.0-ге дейін жасайды

randint (): кездейсоқ санды белгілі бір ауқымнан қайтарады

randrange (): белгілі бір сандар жиынтығынан кездейсоқ санды қайтарады

shuffle (): тізімді араластырады

choice (): кездейсоқ тізім элементін қайтарады

Random () функциясы 0.0-ден 1.0-ге дейінгі аралықта өзгермелі нүктесі бар кездейсоқ санды қайтарады. Егер бізге үлкен ауқымның саны қажет болса, 0-ден 100-ге дейін айтайық, сәйкесінше random функциясының нәтижесін 100-ге көбейте аламыз.  
import random

number = random.random()  # значение от 0.0 до 1.0

print(number)

number = random.random() \* 100  # значение от 0.0 до 100.0

print(number)

Randint(min, max) функциясы кездейсоқ бүтін санды екі min және max мәндерінің арасындағы бос орынға қайтарады.

import random

number = random.randint(20, 35)  # значение от 20 до 35

print(number)

**Python-да өз модулін құру**

Енді модульді құру уақыты келді. Файлды жасаңыз mymodule.py, онда біз кейбір функцияларды анықтаймыз:

**def** hello():

print('Hello, world!')

**def** fib(n):

a = b = 1

**for** i **in** range(n - 2):

a, b = b, a + b

**return** b

Енді сол бумада басқа файл жасаңыз, мысалы main.py:

**import** **mymodule**

mymodule.hello()

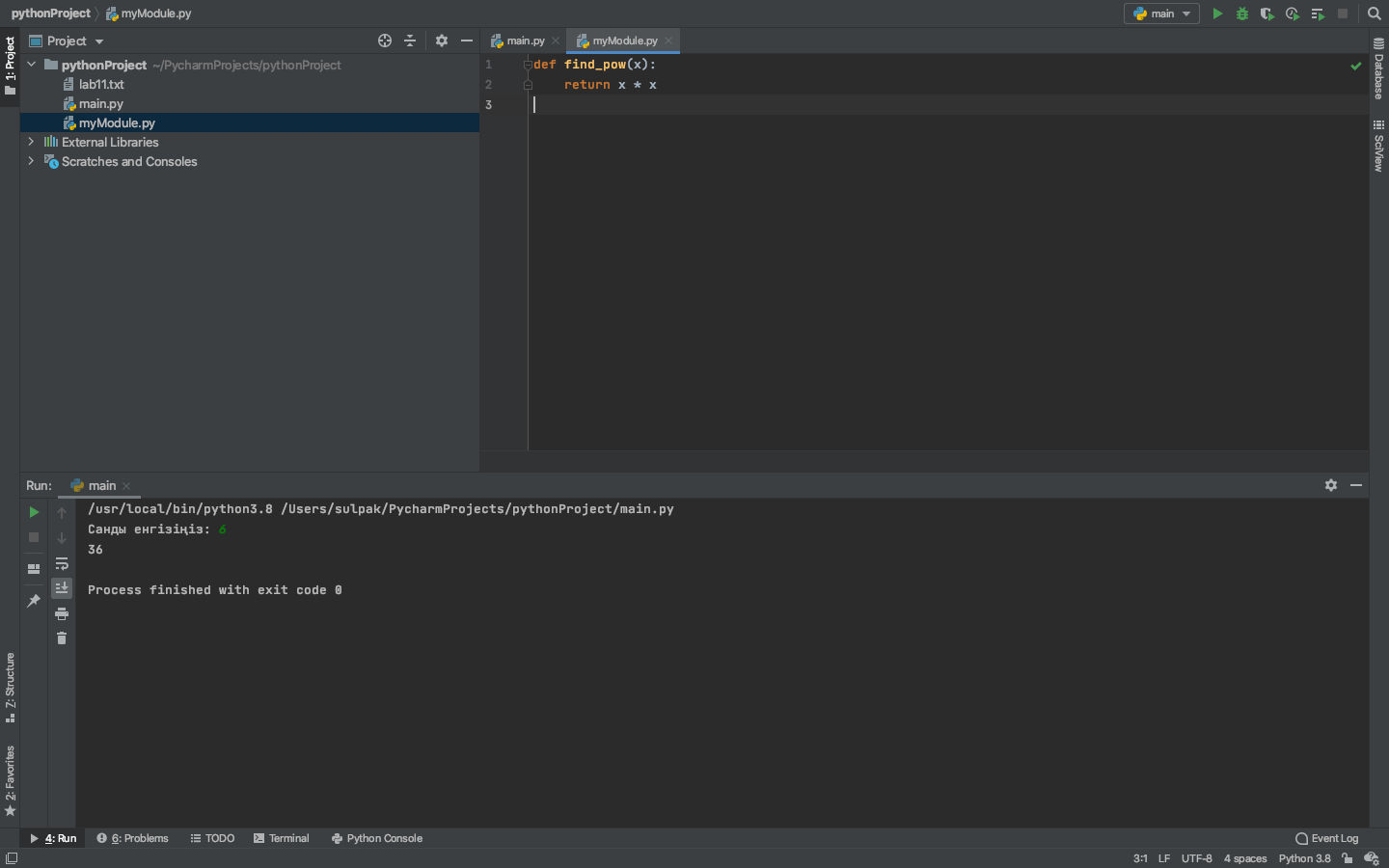
print(mymodule.fib(10))

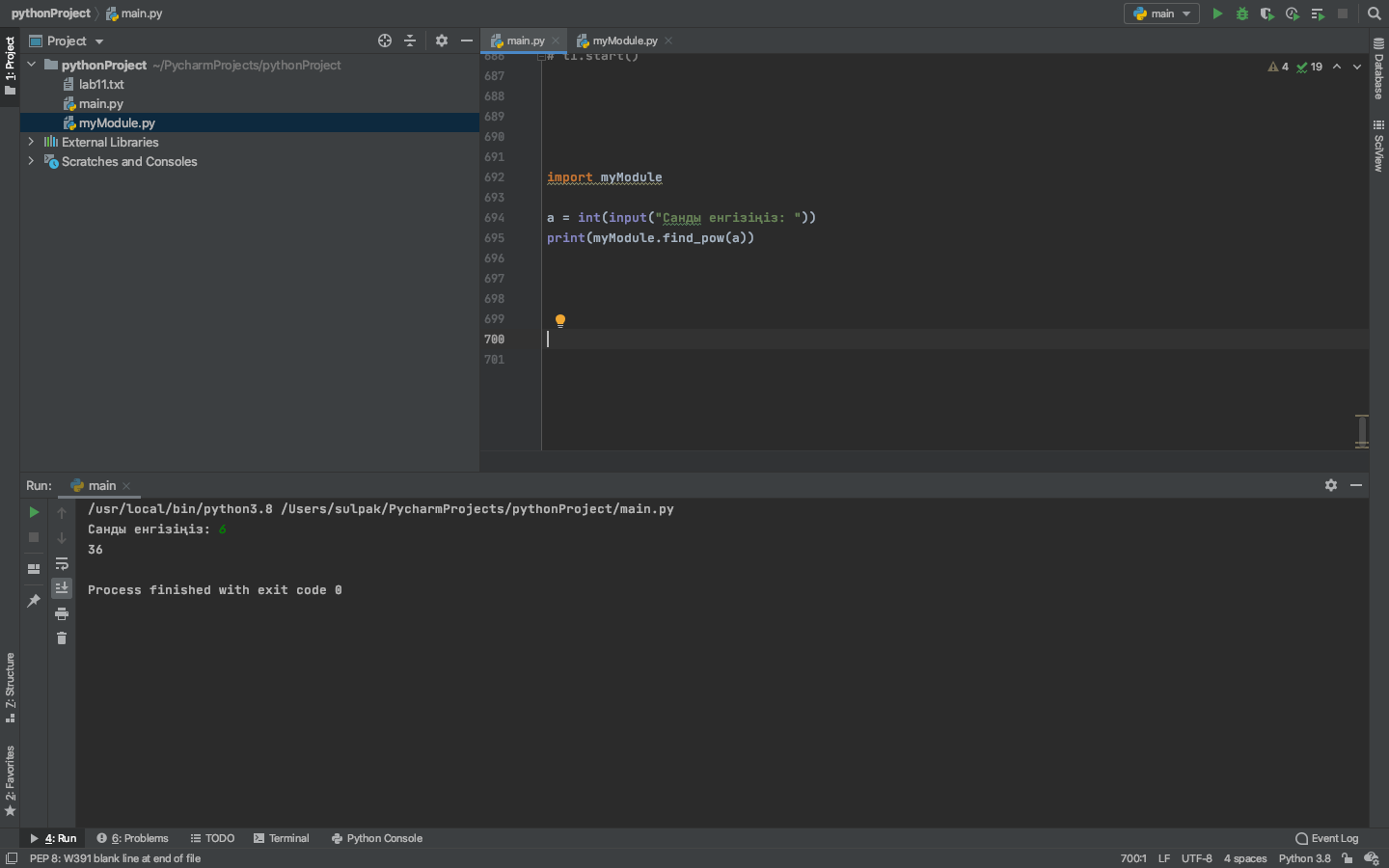
Шығатыны:

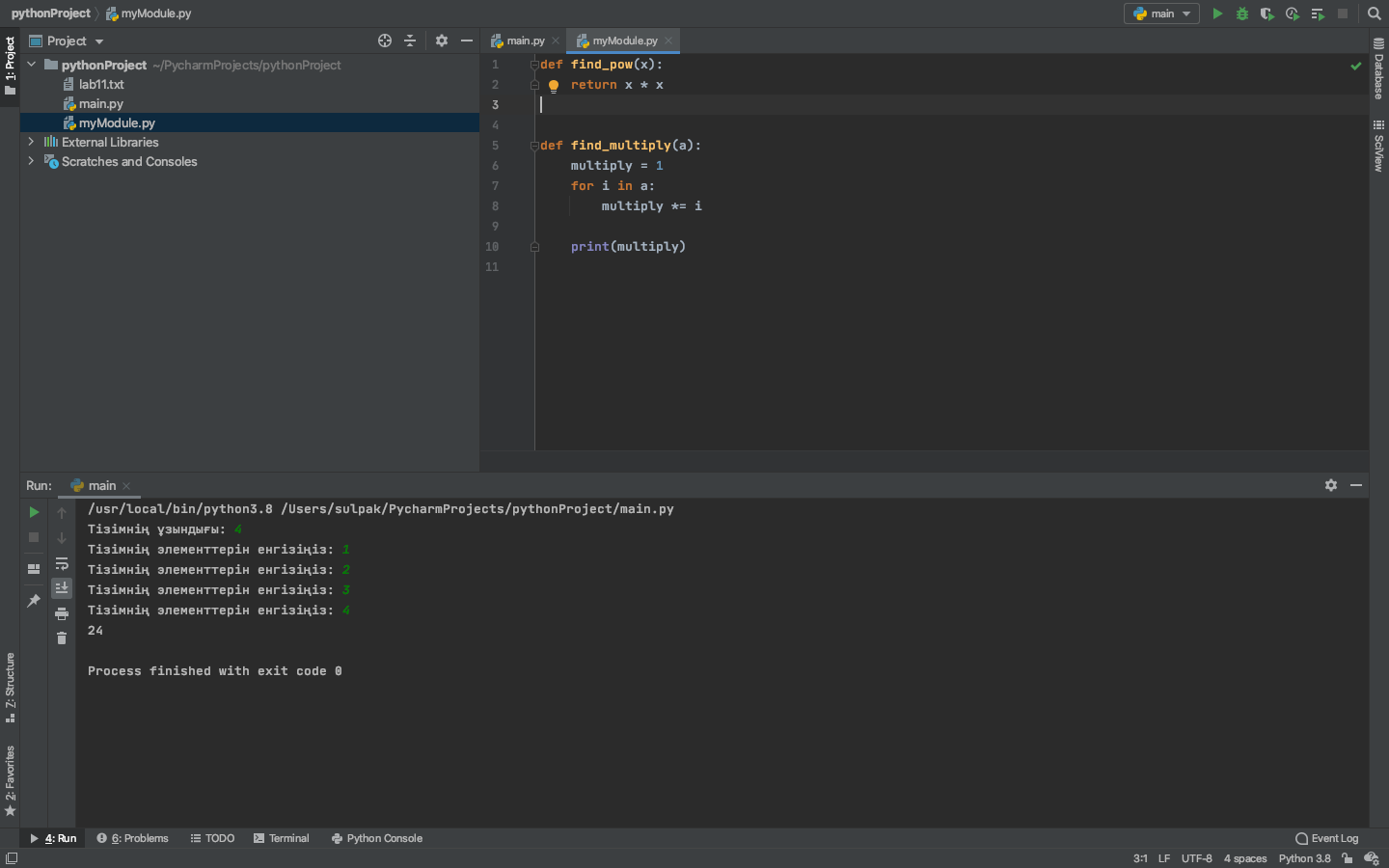
Hello, world!

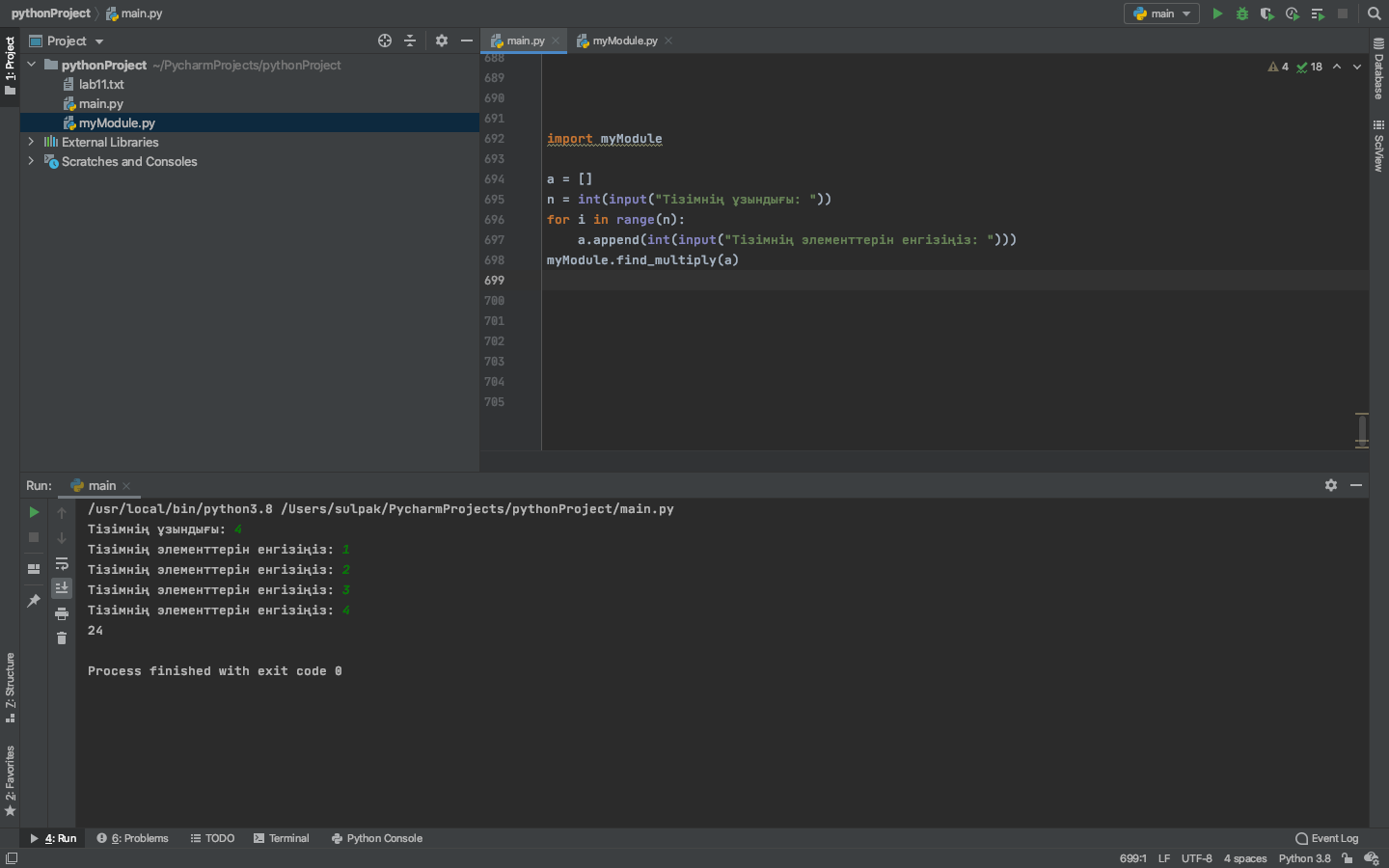
55

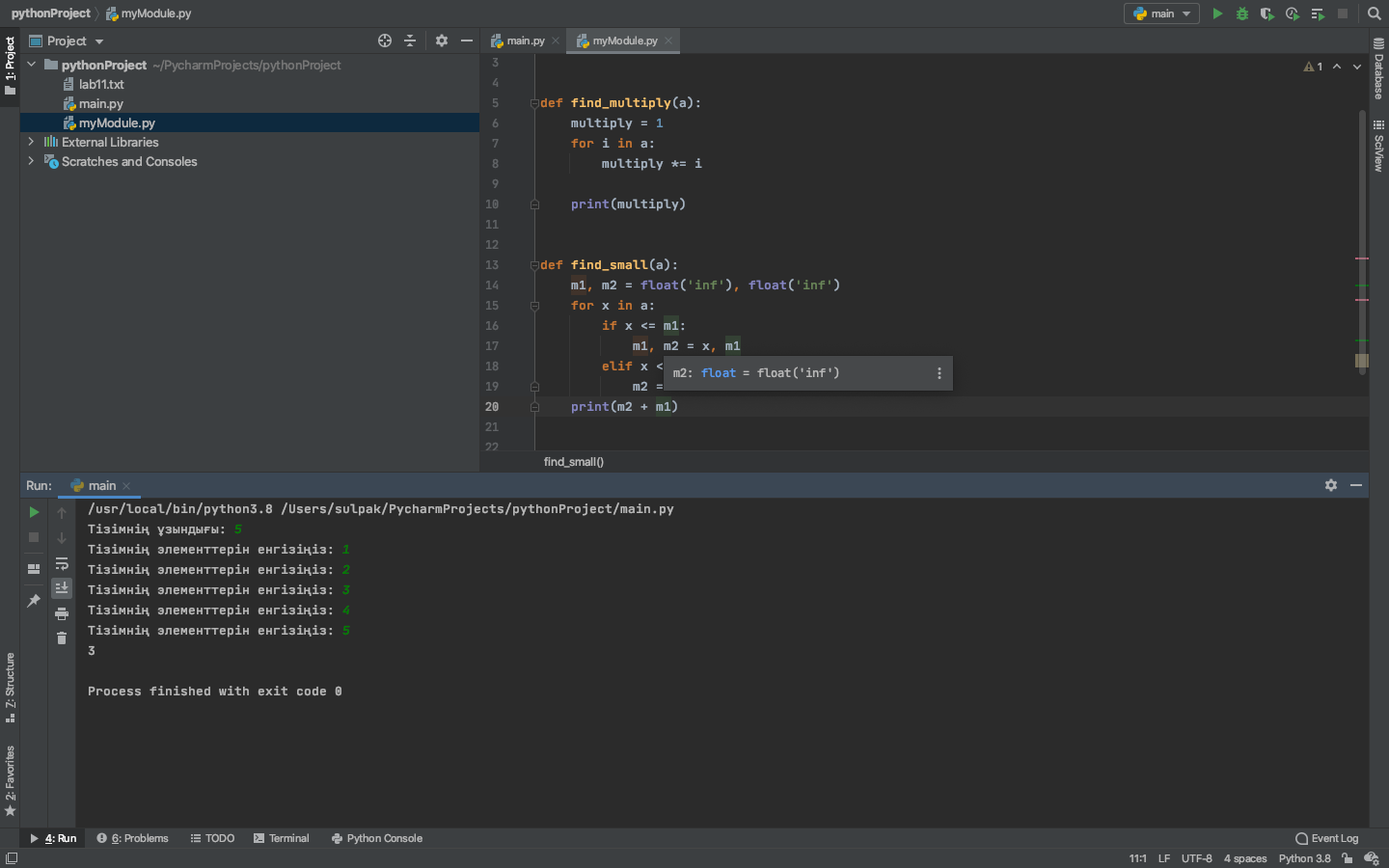
**Тапсырмалар:**

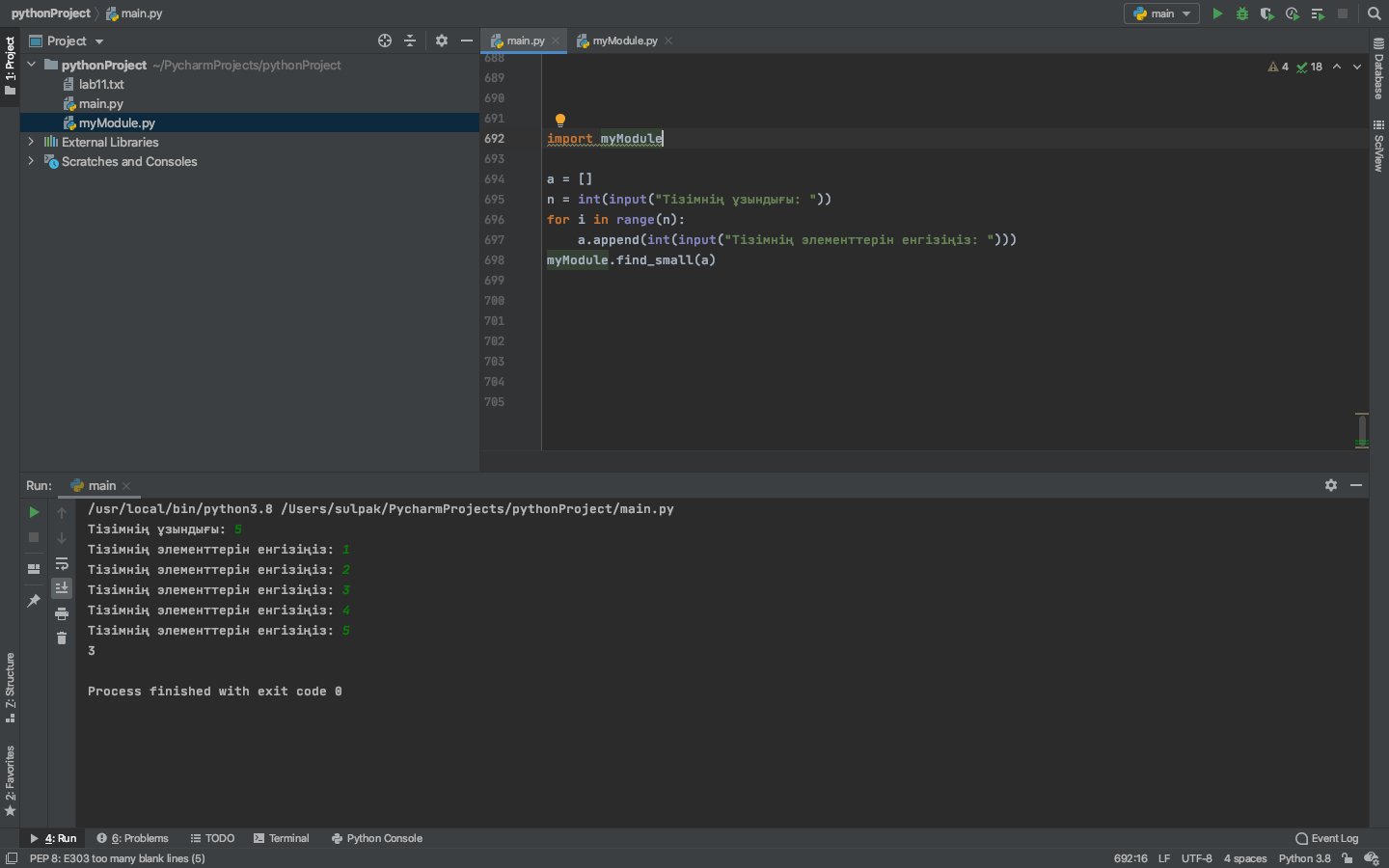
1. Тақырып бойынша әдебиеттерден ақпаратты зерттеу;
2. Python модульдеріне сипаттама беру.
3. Python-да өз модулін құру



1. А тізімінде N саны бар. Тізім элементтерінің мәндерінің көбейтіндісін табыңыз. Функция мақсаты: тізім элементтерінің көбейтіндісін есептеу. Құрылған функцияны бағдарламалық модуль түрінде жасаңыз және оны негізгі бағдарламаға қосыңыз.



1. А тізімінде N саны бар. Берілген мәннен кіші тізім элементтерінің мәндерінің қосындысын табыңыз. Функцияның мақсаты: мәні берілгеннен аз элементтерді жинақтау. Құрылған функцияны бағдарламалық модуль түрінде жасаңыз және оны негізгі бағдарламаға қосыңыз.



**Бақылау сұрақтары:**

1. Бағдарламалардың модульдік құрылымының ерекшеліктерін ашыңыз. Бағдарламалаудың осы әдісінің артықшылықтарын негіздеңіз.
2. Python тілінде модульдерді құрудың ауызша алгоритмін сипаттаңыз.
3. Жасалған пайдаланушы модулін Python тілінде жазылған бағдарламаға қалай қосуға болады?
4. Модуль қалай құрылады?
5. from және import нұсқаулары қалай байланысты?
6. Модульдің бастапқы коды бар файл модульдік нысанға қалай айналады?
7. Майк Мак Грат. Программирование на Python для начинающих: [перевод с англ. М.А. Райтмана]/ Майк Мак Грат. - Москва: Эксмо, 2015.- 192с./<https://www.litmir.me/bd/?b=608860&p=1>
8. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2017. — 832 с.: ил. <http://i.uran.ru/webcab/system/files/bookspdf/python-3-i-pyqt-5-razrabotka-prilozheniy/python.pdf>
9. <https://youtu.be/bTjvm7Zz2wU>
10. Модуль Python <https://youtu.be/d-o5BZA56oI>