

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №4

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Мамонтов Д.В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

ХОД РАБОТЫ

```
>>> print('Hello world!')
Hello world!
```

Рисунок 1 – проверка работоспособности

```
base) C:\Users\TEPLOX0>jupyter notebook
I 17:13:31.942 NotebookApp] Writing notebook server cookie secret to C:\Users\TEPLOX0\AppData\Roaming\jupyter\
tebook_cookie_secret
I 2022-01-23 17:13:32.928 LabApp] JupyterLab extension loaded from C:\Users\TEPLOX0\anaconda3\lib\site-pack
ab
I 2022-01-23 17:13:32.929 LabApp] JupyterLab application directory is C:\Users\TEPLOX0\anaconda3\share\jupy
I 17:13:32.936 NotebookApp] Serving notebooks from local directory: C:\Users\TEPLOX0
I 17:13:32.936 NotebookApp] Jupyter Notebook 6.4.5 is running at:
I 17:13:32.936 NotebookApp] http://localhost:8888/?token=475938816a6fb3470de6d7edeea6da8851b9de3eef7a1433
I 17:13:32.936 NotebookApp] or http://127.0.0.1:8888/?token=475938816a6fb3470de6d7edeea6da8851b9de3eef7a14
I 17:13:32.936 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip conf
C 17:13:32.975 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/TEPLOX0/AppData/Roaming/jupyter/runtime/nbserver-12256-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=475938816a6fb3470de6d7edeea6da8851b9de3eef7a1433
    or http://127.0.0.1:8888/?token=475938816a6fb3470de6d7edeea6da8851b9de3eef7a1433
```

Ввод [1]: `print('Hello world')`

Hello world

Рисунок 2 и 3 – проверка anaconda

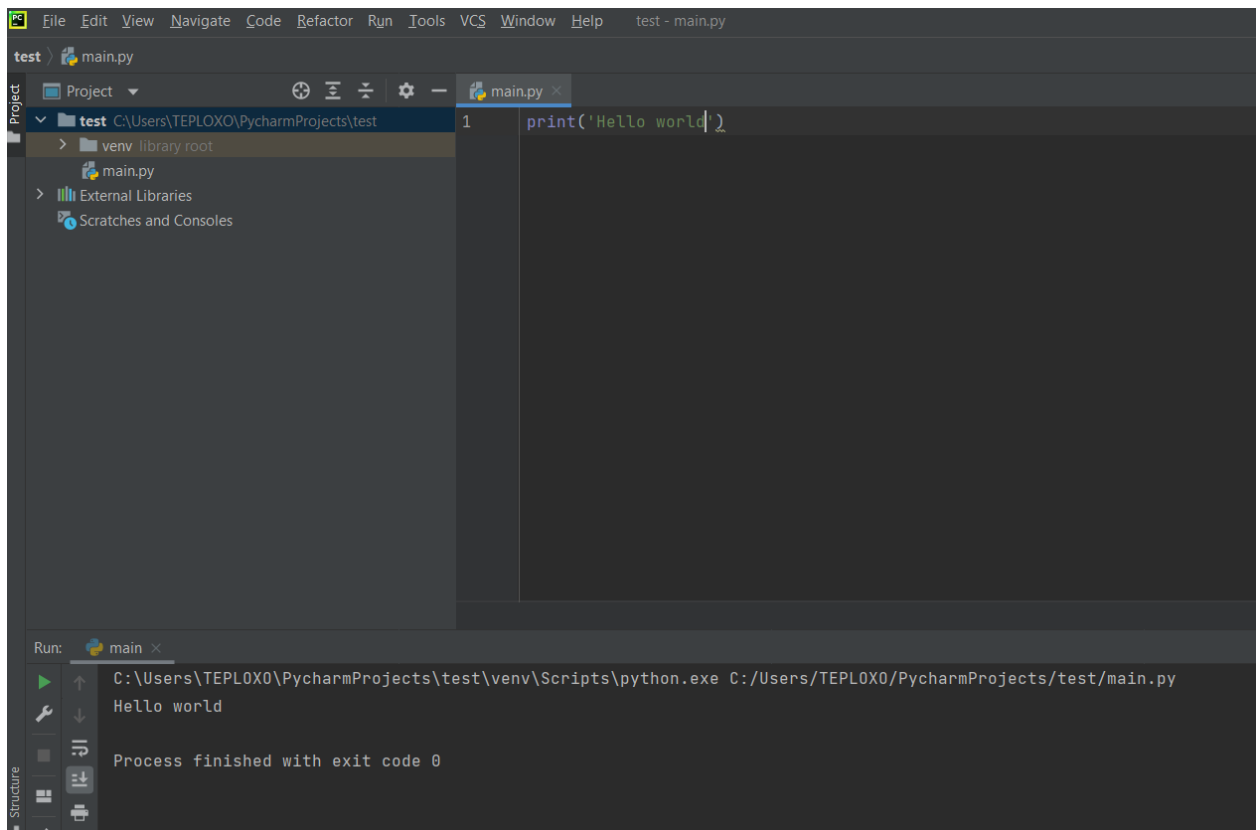


Рисунок 4 – проверка pycharm

```
(base) C:\Users\TEPLOX0>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Рисунок 5 – запуск python в интерактивном режиме

```
(base) C:\Users\TEPLOX0\PycharmProjects\test>python test.py
2
3
```

Рисунок 6 – запуск python в пакетном режиме

```
>>> print("Python keywords: ", keyword.kwlist)
Python keywords: ['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'l
ambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Рисунок 7 – просмотр ключевых слов

```
>>> keyword.iskeyword("try")
True
>>> keyword.iskeyword("b")
False
```

Рисунок 8 – проверка идентификаторов на причастие к ключевым словам

```
>>> a=4
>>> b=5
>>> id(a)
1966858201488
>>> id(b)
1966858201520
>>> a=b
>>> id(a)
1966858201520
```

Рисунок 9 – просмотр идентификатора объекта

```
>>> a=10
>>> b="hello"
>>> c=(1,2)
>>> type(a)
<class 'int'>
>>> type(b)
<class 'str'>
>>> type(c)
<class 'tuple'>
```

Рисунок 10 – просмотр типов переменных

```
>>> k=15
>>> id(k)
1966858201840
>>> type(k)
<class 'int'>
```

Рисунок 11 – неизменяемый тип данных

```
>>> a=[1,2]
>>> id(a)
1966894169152
>>> a[1]=3
>>> a
[1, 3]
>>> id(a)
1966894169152
```

Рисунок 12 – изменяемый тип данных

```
>>> 3+2
5
>>> a=3
>>> b=2
>>> a+b
5
>>> a=3
>>> b=2
>>> c=a+b
>>> print(c)
5
>>> a=a+b
>>> print(a)
5
>>> a=3
>>> a+=b
>>> print(a)
5
```

Рисунок 13 – арифметические операции и их сокращения

```
>>> 4-2
2
>>> a=5
>>> b=7
>>> a-b
-2
```

Рисунок 14 – операция вычитания

```
>>> 5*8
40
>>> a=4
>>> a*=10
>>> print(a)
40
```

Рисунок 15 – операция умножения

```
>>> 9/3
3.0
>>> a=7
>>> b=4
>>> a/b
1.75
```

Рисунок 16 – операция деления

```
>>> 9//3
3
>>> a=7
>>> b=4
>>> a//b
1
```

Рисунок 17 – получение целой части от деления

```
>>> 9%5
4
>>> a%b
3
```

Рисунок 18 – получение остатка от деления

```
>>> 5**4
625
>>> a**b
2401
```

Рисунок 19 – возведение в степень

```
>>> z=1+2j
>>> print(z)
(1+2j)
>>> x=complex(3,2)
>>> print(x)
(3+2j)
>>> x+z
(4+4j)
>>> x-z
(2+0j)
>>> x*z
(-1+8j)
>>> x/z
(1.4-0.8j)
>>> x**z
(-1.1122722036363393-0.012635185355335208j)
>>> x**3
(-9+46j)
>>> x=3+2j
>>> x.real
3.0
>>> x.imag
2.0
>>> x.conjugate()
(3-2j)
```

Рисунок 20 – комплексные числа и арифметические операции с ними, а также извлечение действительной и мнимой части

```

>>> p=9
>>> q=3
>>> p&q
1
>>> p|q
11
>>> p^q
10
>>> ~p
-10
>>> p<<1
18
>>> p>>1
4

```

Рисунок 21 – битовые операции

```

>>> m=124504
>>> hex(m)
'0x1e658'
>>> oct(m)
'0o363130'
>>> bin(m)
'0b11110011001011000'

```

Рисунок 22 – представление числе в других системах счисления

```

>>> import math
>>> math.ceil(3.2)
4
>>> math.fabs(-7)
7.0
>>> math.factorial(5)
120
>>> math.floor(3.2)
3
>>> math.exp(3)
20.085536923187668
>>> math.log2(2)
1.0
>>> math.log10(2)
0.3010299956639812
>>> math.log(4, 8)
0.6666666666666667
>>> math.pow(3,4)
81.0
>>> math.sqrt(25)
5.0
>>> math.cos(0)
1.0
>>> math.sin(math.pi/6)
0.49999999999999994
>>> math.tan(math.pi/4)
0.9999999999999999
>>> math.acos(-1)
3.141592653589793

```

```
>>> math.acos(-1)
3.141592653589793
>>> math.asin(1)
1.5707963267948966
>>> math.atan(0.5)
0.4636476090008061
>>> math.pi
3.141592653589793
>>> math.e
2.718281828459045
```

Рисунки 23 и 24 – математические функции модуля math

```
>>> print(1032)
1032
>>> print(2.34)
2.34
>>> print("Hello")
Hello
>>> print("a:",1)
a: 1
>>> one=1
>>> two=2
>>> three=3
>>> print(one, two, three)
1 2 3
>>> print("hello" + " " + "world")
hello world
>>> print(10-2.5/2)
8.75
>>> print("Mon","Tue","Wed","Thu","Fri","Sat","Sun", sep="-")
Mon-Tue-Wed-Thu-Fri-Sat-Sun
>>> print(1,2,3, sep="//")
1//2//3
>>> print(10,end="")
10>>> print(10, end='\n')
10
>>> print(10, end='\n\n')
10
```

```
>>> pupil="Ben"
>>> old=16
>>> grade=9.2
>>> print("It`s %s, %d. Level: %f" %(pupil,old,grade))
It`s Ben, 16. Level: 9.200000
>>> print("It`s %s, %d. Level: %.1f" %(pupil,old,grade))
It`s Ben, 16. Level: 9.2
```

```
>>> print("This is a {0}. It`s {1}." .format("ball","red"))
This is a ball. It`s red.
>>> print("This is a {0}. It`s {1}." .format("cat","white"))
This is a cat. It`s white.
>>> print("This is a {0}. It`s {1} {2}." .format(1,"a","number"))
This is a 1. It`s a number.
```

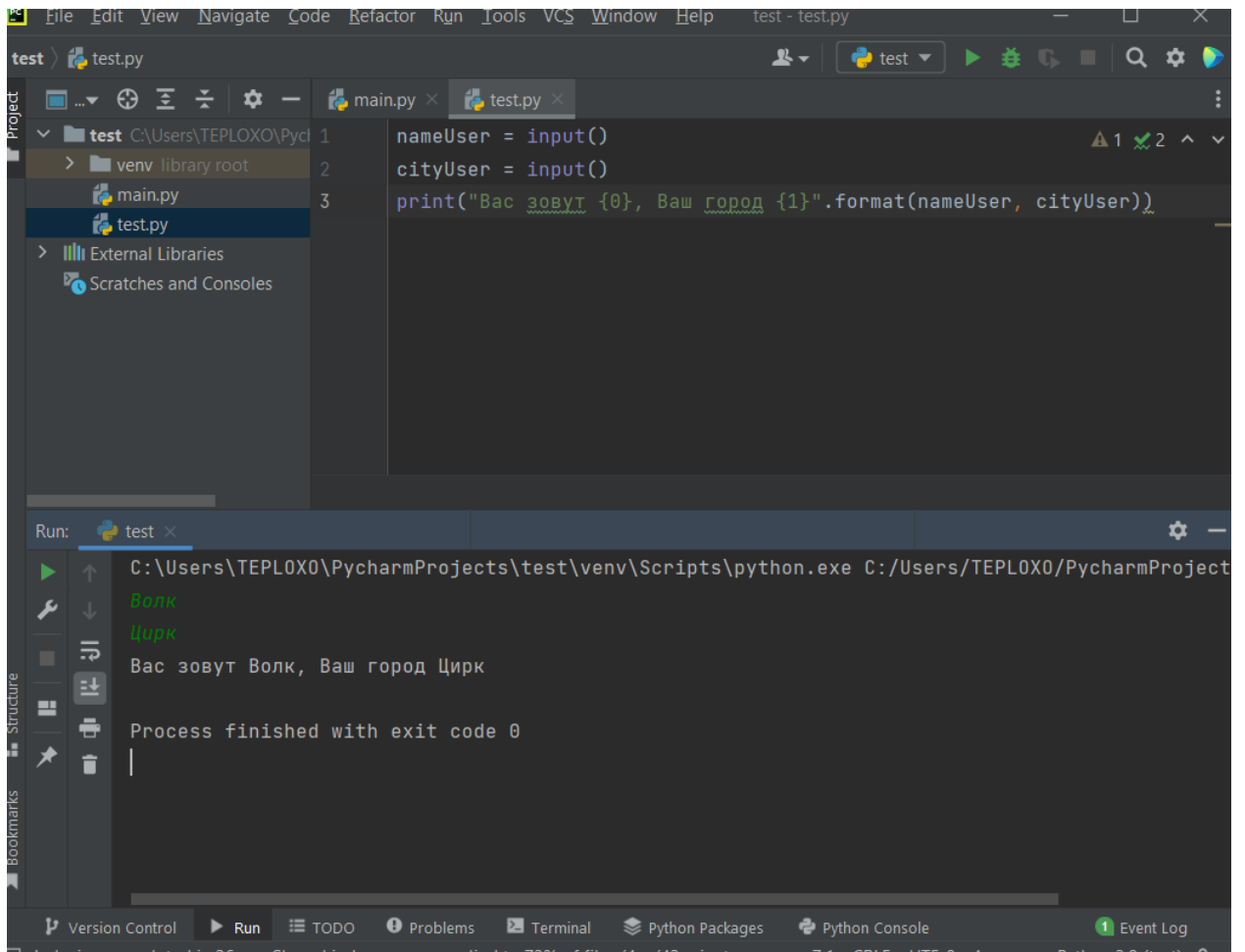
Рисунки 25,26 и 27 – функции вывода print

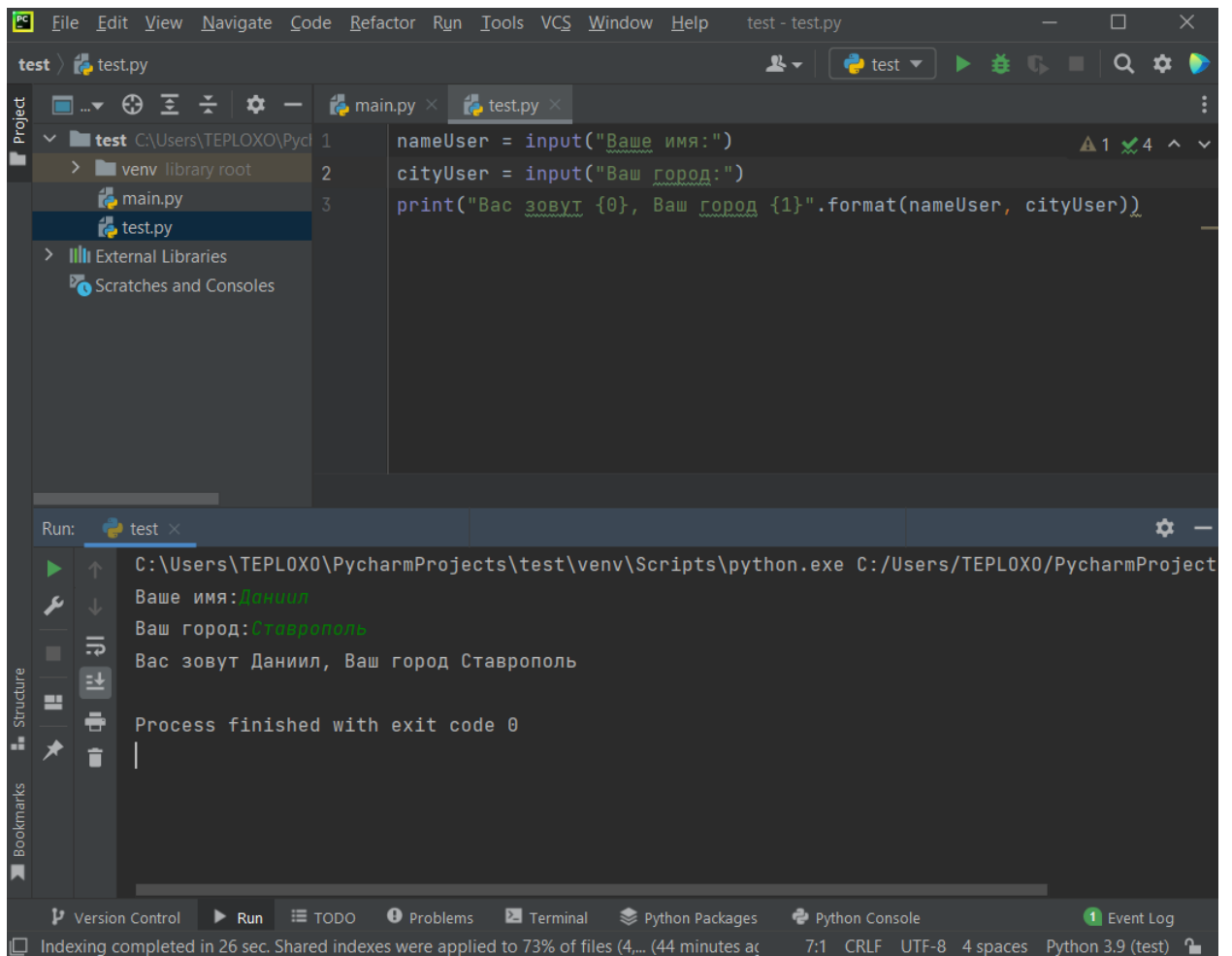
```

>>> input()
Yes!
'Yes!'
>>> answer=input()
No, it is not.
>>> answer
'No, it is not.'
>>> print(answer)
No, it is not.

```

Рисунок 28 – ввод данных при помощи функции input





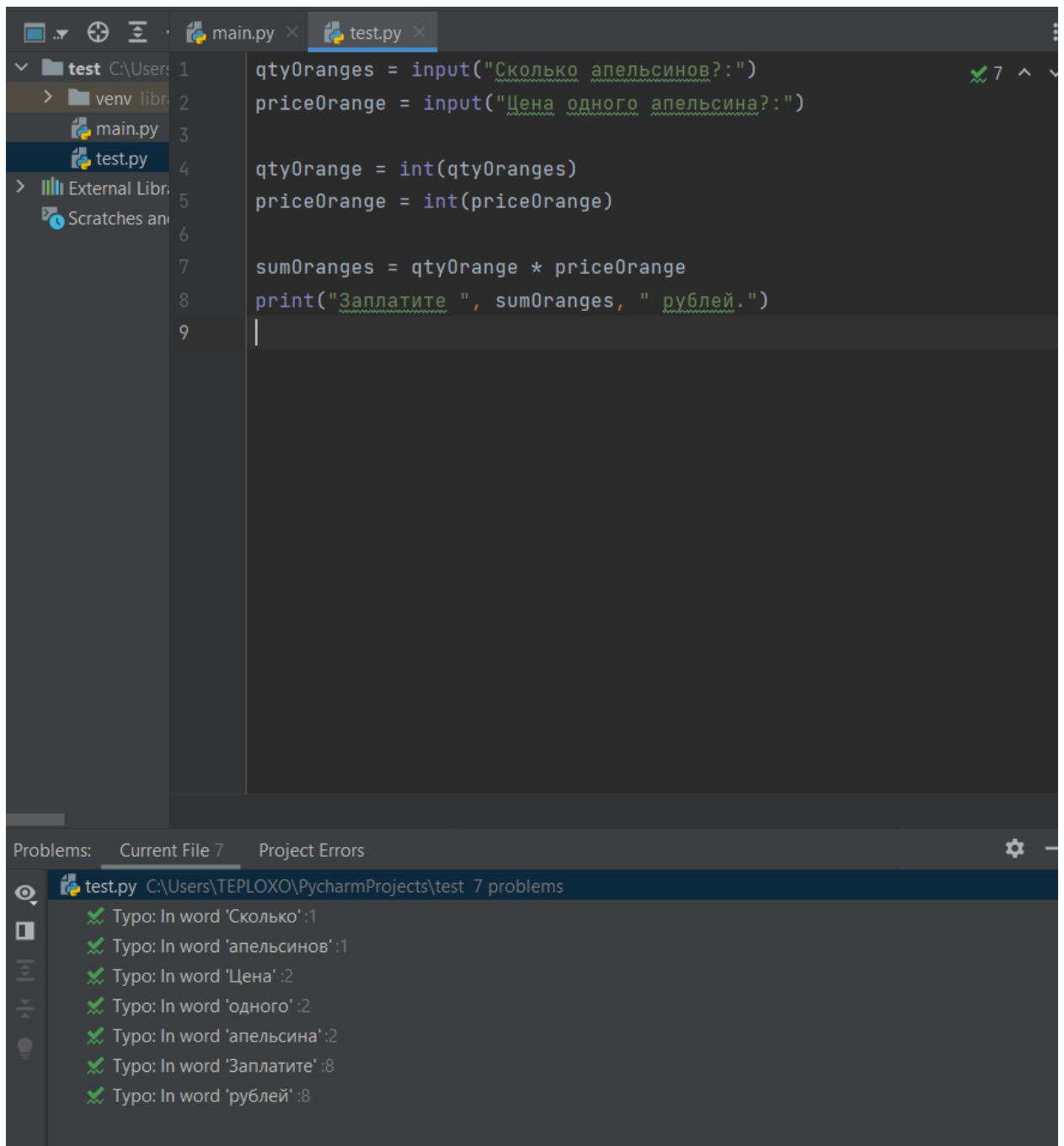


Рисунок 29,30 и 31 – input в скриптах

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Известна стоимость 1 кг конфет, печенья и яблок. Найти стоимость всей покупки, если купили кг конфет, у кг печенья и кг яблок.

```
priceSweets = int(input("Конфеты, цена за кг.:"))
priceCookie = int(input("Печенье, цена за кг.:"))
priceApple = int(input("Яблоки, цена за кг.:"))

amountSweets = int(input("Сколько кг. конфет купили?:"))
amountCookie = int(input("Сколько кг. печенья купили?:"))
amountApple = int(input("Сколько кг. яблок купили?:"))

allSum = (amountSweets * priceSweets) + (amountCookie * priceCookie) +
(amountApple * priceApple)
print("Заплатите ", allSum, " рублей.")
```

```
Конфеты, цена за кг.:30
Печенье, цена за кг.:15
Яблоки, цена за кг.:10
Сколько кг. конфет купили?:1
Сколько кг. печенья купили?:2
Сколько кг. яблок купили?:4
Заплатите 100 рублей.

Process finished with exit code 0
```

Рисунок 1 – результат расчетов

ЗАДАНИЕ ПОВЫШЕННОЙ СЛОЖНОСТИ

Даны цифры двух десятичных целых чисел: трехзначного и двузначного, где a – число единиц, b – число десятков, c – число сотен. Получить цифры числа, равного сумме заданных чисел (известно, что это число трехзначное). Числа-слагаемые и число-результат не определять; условный оператор не использовать.

```
a = (int(input('a1=')) + 10 * int(input('a2=')) + 100 * int(input('a3=')))
b = (int(input('b1=')) + 10 * int(input('b2=')))
print(a + b)
```

```
a1 = 3
a2 = 2
a3 = 1
b1 = 4
b2 = 2
147
```

Рисунок 1 – результат выполнения кода

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Запустить скачанный установочный файл.
Выбрать способ установки.
Отметить необходимые опции установки.
Выбрать место установки.
2. Anaconda уже включает в себя интерпретатор языка python, набор наиболее используемых библиотек и удобную среду разработки, запускаемую в браузере.
3. Для проверки работоспособности Anaconda необходимо вначале запустить консоль с поддержкой виртуальных окружений Anaconda. В появившемся окне ввести jupyter notebook, в результате чего отобразится процесс загрузки веб среды. Затем запустится веб-сервер и среда разработки в браузере. Создайте новый проект python в jupyter notebook и затем введите команду `print("Hello world")`. Если команда была исполнена, значит все в порядке.
4. При создании нового проекта, pycharm предложит выбрать желаемый интерпретатор самостоятельно.
5. Создайте новый проект в PyCharm, добавьте python файл в проект, введите код программы, запустите программу с помощью `shift+10`. Либо создайте текстовый документ с расширением .py, нажмите на него правой кнопкой мыши и выберите опцию открыть при помощи PyCharm.
6. В интерактивном режиме вы можете сразу начать создавать исходный код. При пакетном режиме вам понадобится сначала создать файл с исходным кодом, который будет использован для осуществления работы созданной вами программы.
7. При объявлении переменной вам не нужно указывать ее тип.
8. К основным типам относятся: none, логические переменные, числа, списки, строки, бинарные списки, множества, словари.
9. В python существуют объекты – они являются абстракцией для представления данных, данные – это числа, списки, строки и т.п. При этом, под данными понимаются непосредственно сами объекты. Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор помогает отличать объекты друг от друга. При инициализации переменной создается целочисленный объект, данный объект имеет некоторый идентификатор, значение 5, и тип: целое число.
10. Чтобы вывести список ключевых слов, нужно подключить модуль keyword и воспользоваться командой `keyword.kwlist`.
11. Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`. Функция `type` показывает тип данных, внесенных в переменную.
12. Неизменяемыми типами являются типы, которые не могут изменяться. Если тип данных изменяемый, то можно менять значение объекта.
13. При обычном делении вы получите число с его мнимой частью. При целочисленном делении же вы получите только целое число.
14. В python входят функции создания комплексных чисел `complex(a,b)`, их сложения, вычитания, умножения, деления, возведения в степень, а также функция извлечения действительной и мнимой части.
15. В библиотеке math содержится большое количество часто используемых математических функций.
16. `sep` ставит отличный от пробела разделитель строк. Параметр `end` позволяет указать, что делать после вывода строки. По умолчанию происходит переход на новую строку.
17. Метод `format()` позволяет подставлять определенные данные в строку.
18. `A=5` – целочисленная переменная, `a=25.5` – вещественная переменная.