МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчёт о лабораторной работе №8 по дисциплине основы программной инженерии

Выполнила:
Нестеренко Тамара Антоновна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ

1. Практическая часть

```
>>> a = [1, 2, 3]
>>> print(a)
[1, 2, 3]
>>> a[1] = 15
>>> print(a)
[1, 15, 3]
>>>
```

Рисунок 1 – Пример замены элементов списка

```
>>> b = (1, 2, 3)
>>> print(b)
(1, 2, 3)
>>> b[1] = 15
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

Рисунок 2 – Пример невозможности замены элемента в кортеже

```
>>> lst = [10, 20, 30]
>>> tpl = (10, 20, 30)
>>> print(lst.__sizeof__())
104
>>> print(tpl.__sizeof__())
48
>>>
```

Рисунок 3 – Пример памяти, занимаемой кортежем и списком

```
>>> a = ()
>>> print (type(a))
<class 'tuple'>
>>> b = tuple()
>>> print(type(b))
<class 'tuple'>
```

Рисунок 4 – Пример создания кортежа

```
>>> a = (1, 2, 3, 4, 5)
>>> print(type(a))
<class 'tuple'>
>>> print(a)
(1, 2, 3, 4, 5)
>>>
```

Рисунок 5 – Пример представления кортежа

```
>>> a = tuple([1, 2, 3, 4])
>>> print(a)
(1, 2, 3, 4)
```

Рисунок 6 – Пример использования функции tuple() для создания кортежа

```
not_a_tople = (42) #42
tuple = (42,) #(42,)
```

Рисунок 7 – Пример кортежа с одним элементом

```
>>> a = (1, 2, 3, 4, 5)
>>> print(a[0])
1
>>> print(a[1:3])
(2, 3)
>>> a[1]
2
>>> a[1] = 3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

Рисунок 8 – Пример доступа к элементам кортежа

```
>>> a = (1, 2, 3, 4, 5)
>>> del a[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object doesn't support item deletion
>>>
```

Рисунок 9 – Пример удаления одного элемента кортежа

```
>>> del a
>>> print (a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>>
```

Рисунок 10 – Пример удаления кортежа полностью

```
>>> lst = [1, 2, 3, 4, 5]
>>> print(type(lst))
<class 'list'>
>>> print(lst)
[1, 2, 3, 4, 5]
>>> tpl = tuple(lst)
>>> print(type(tpl))
<class 'tuple'>
>>> print(tpl)
(1, 2, 3, 4, 5)
>>>
```

Рисунок 11 – Пример преобразования списка в кортеж

```
>>> tpl = (2, 4, 6, 8, 10)
>>> print(type(tpl))
<class 'tuple'>
>>> print(tpl)
(2, 4, 6, 8, 10)
>>> lst = list(tpl)
>>> print(type(lst))
<class 'list'>
>>> print(lst)
[2, 4, 6, 8, 10]
>>>
```

Рисунок 12 – Пример преобразования кортежа в список

```
name_and_age = ('Bob', 42)

(name, age) = name_and_age
name # 'Bob'
age  # 42
```

Рисунок 13 – Пример деструктуризации кортежа

Рисунок 14 – Пример способа, который получает и сразу разбирает значение функции

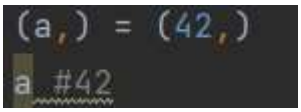

Рисунок 15 – Пример разбора кортежа с одним элементом

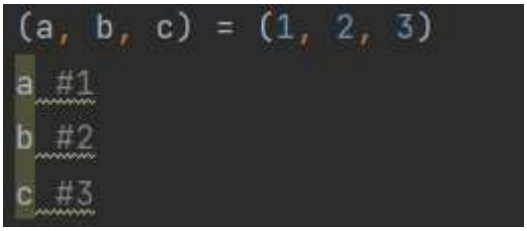

Рисунок 16 – Пример множественно присваивания значения элементам кортежа

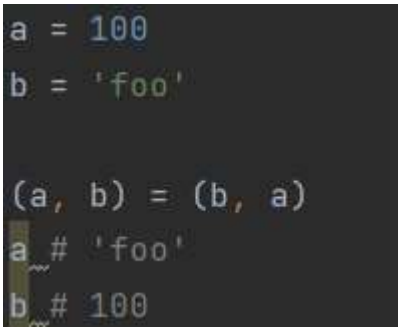

Рисунок 17 – Пример обмена значениями между двумя переменными

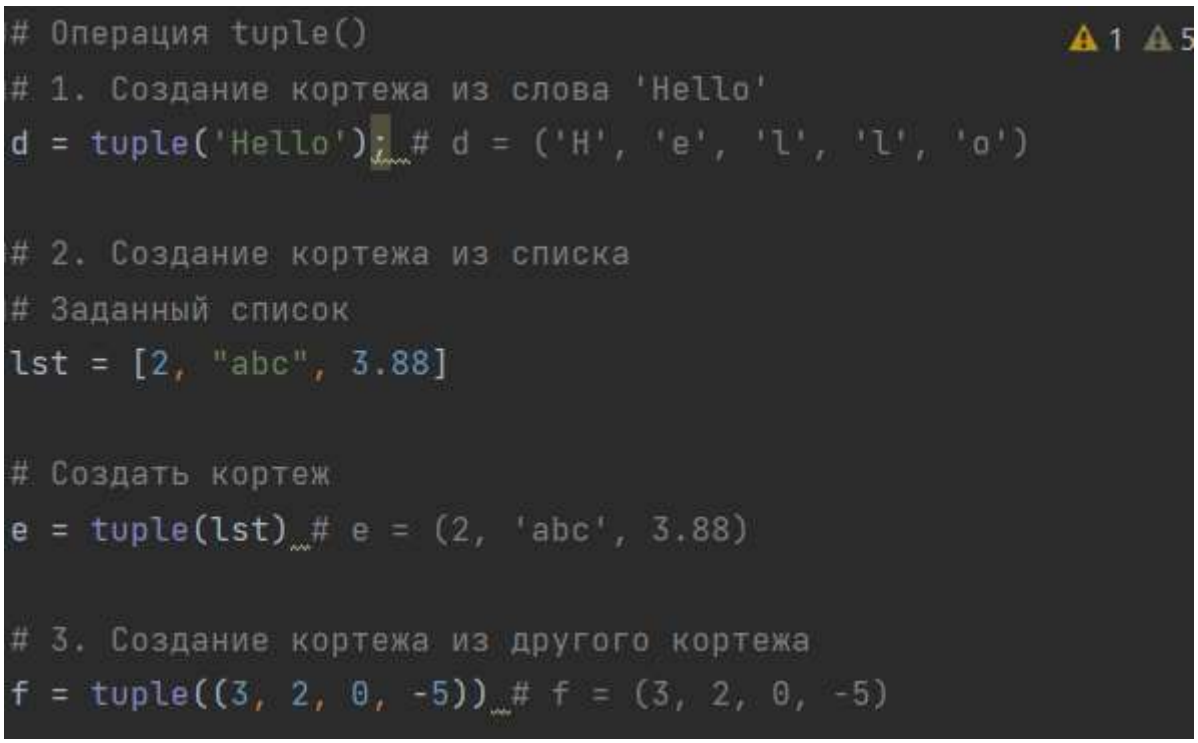

Рисунок 18 – Пример создания кортежа из итерированного объекта

```
# Операция [i:j] - взятие среза
# 1. Кортеж, содержащий целые числа
A = (0, 1, 2, 3)
item = A[0:2] # item = (0, 1)

# 2. Кортеж, содержащий список
A = (2.5, ['abcd', True, 3.1415], 8, False, 'z')
item = A[1:3] # item = (['abcd', True, 3.1415], 8)

# 3. Кортеж, содержащий вложенный кортеж
A = (3, 8, -11, "program")
B = ("Python", A, True)
item = B[:3] # item = ('Python', (3, 8, -11, 'program'), True)
item = B[1:] # item = ((3, 8, -11, 'program'), True)
```

Рисунок 19 – Пример работы операции T[i:j]. Взятие среза в кортеже

```
# Кортежи. Конкатенация +
# Конкатенация двух кортежей
A = (1, 2, 3)
B = (4, 5, 6)
C = A + B # C = (1, 2, 3, 4, 5, 6)

# Конкатенация кортежей со сложными объектам
D = (3, "abc") + (-7.22, ['a', 5]) # D = (3, 'abc', -7.22, ['a', 5])

# Конкатенация трёх кортежей
A = ('a', 'aa', 'aaa')
B = A + (1, 2) + (True, False) # B = ('a', 'aa', 'aaa', 1, 2, True, False)
```

Рисунок 20 – Пример конкатенации кортежей

```
# Кортежи. Повторение *
# Кортеж, который содержит простые числа
A = (1, 2, 3) * 3 # A = (1, 2, 3, 1, 2, 3, 1, 2, 3)

# Кортеж, который содержит вложенные объекты
B = ("ab", ["1", "12"])*2 # A=('ab', ['1', '12'], 'ab', ['1', '12'])
```

Рисунок 21 – Пример повторения кортежа

```python
# Обход кортежа в цикле
# 1. Цикл for
# Заданный кортеж
A = ("abc", "abcd", "bcd", "cde")

# Вывести все элементы кортежа
for item in A:
    print(item)

# 2. Цикл while
# Исходный кортеж - целые числа
A = (-1, 3, -8, 12, -20)

# Вычислить количество положительных чисел
i = 0
k = 0 # количество положительных чисел

while i < len(A):
    if (A[i]<0):
        k = k +1
    i = i +1

# Вывести результат
print("k = ", k)
# 3. Обход в цикле for
# Заданный кортеж, содержащий строки
A = ("abc", "ad", "bcd")

# Сформировать новый список из элементов кортежа A
# в новом списке B, каждый элемент удваивается
B = [item * 2 for item in A]
```

Рисунок 22 – Примеры обхода кортежа в цикле

```
abc
abcd
bcd
cde
k =  3
A =  ('abc', 'ad', 'bcd')
B =  ['abcabc', 'adad', 'bcdbcd']
```

Рисунок 23 – Результат выполнения программы

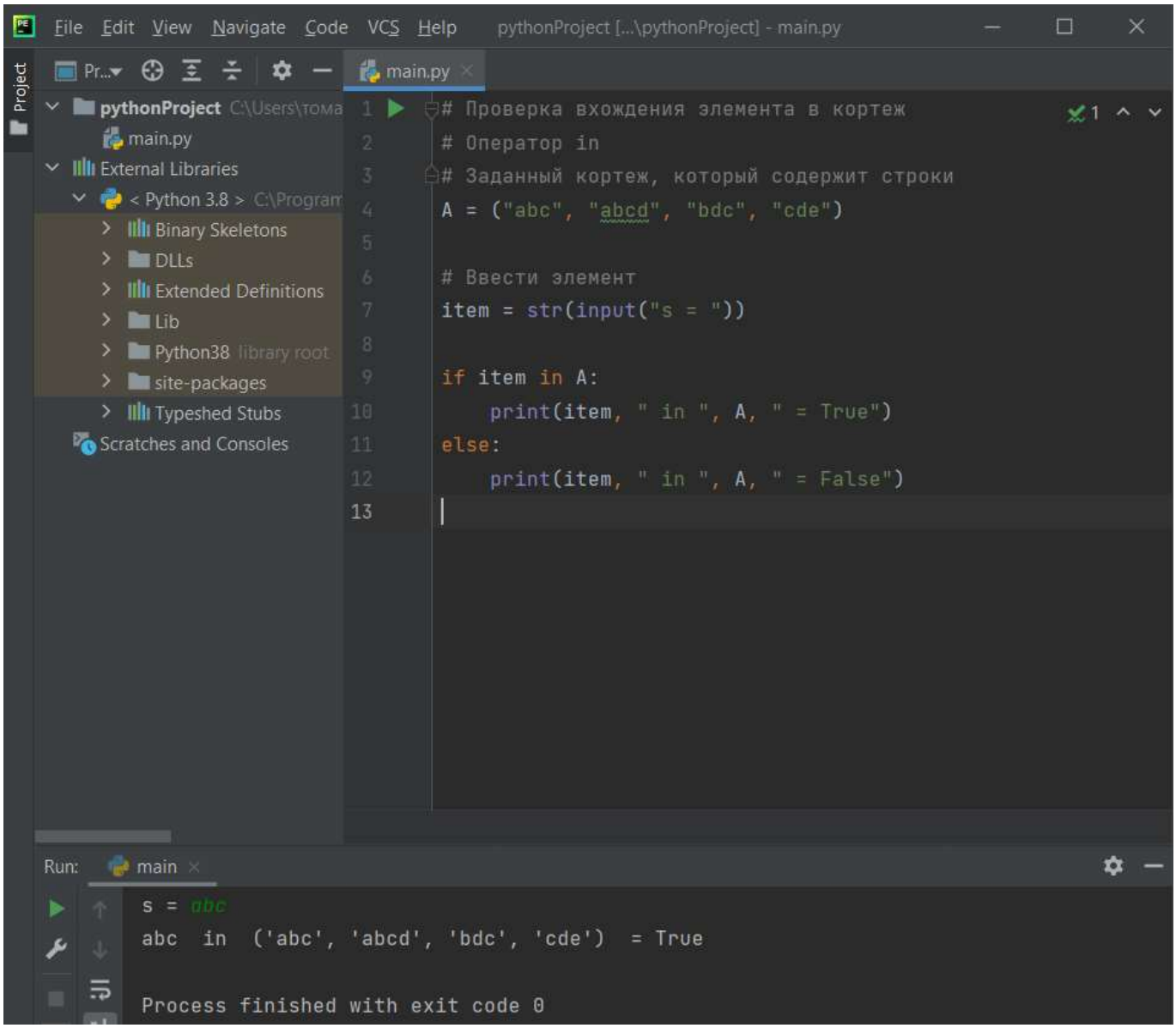

Рисунок 24 – Пример операции in. Проверка вхождения элемента в кортеж

```python
# Метод index - определяет позиции (индекс) элемента в кортеже
# Заданный кортеж
A = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")

# Запрос к вводу названия дня недели
day = str(input("Enter day: "))

# Корректно вычислить индекс
if day in A: # проверка, есть ли строка day в кортеже A
    num = A.index(day)
    print("Number of day = ", num + 1)
else:
    num = -1
    print("Wrong day.")
```

Рисунок 25 – Пример работы метода index(). Поиск позиции элемента в кортеже

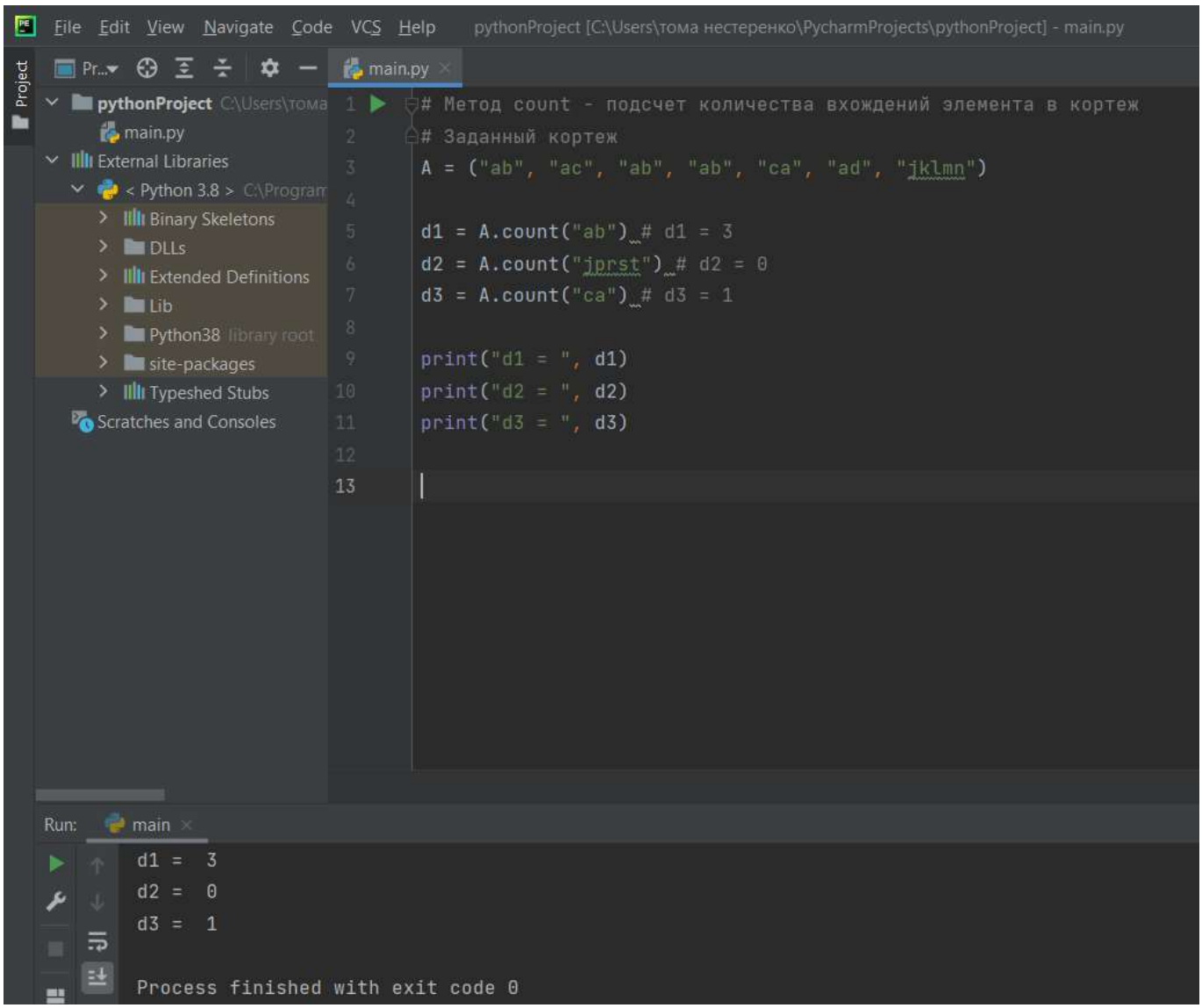

Рисунок 26 – Пример работы метода count(). Количество вхождений элемента в кортеж

2. Работа с примерами

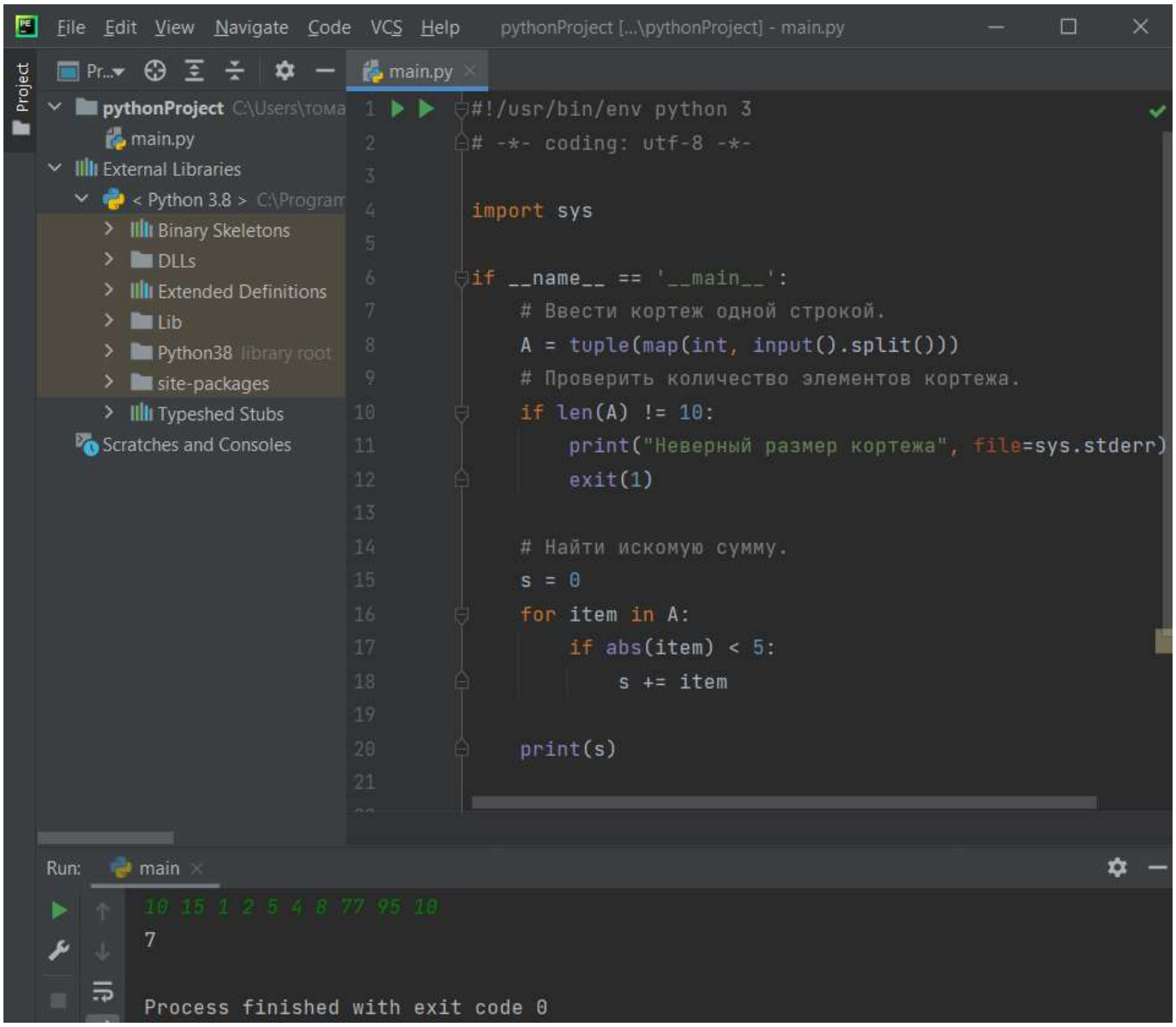

Рисунок 27 – Пример решения задачи: ввести кортеж A из 10 элементов, найти сумму элементов, меньших модулю 5, и вывести её на экран. Использовать в программе вместо списков кортежи.

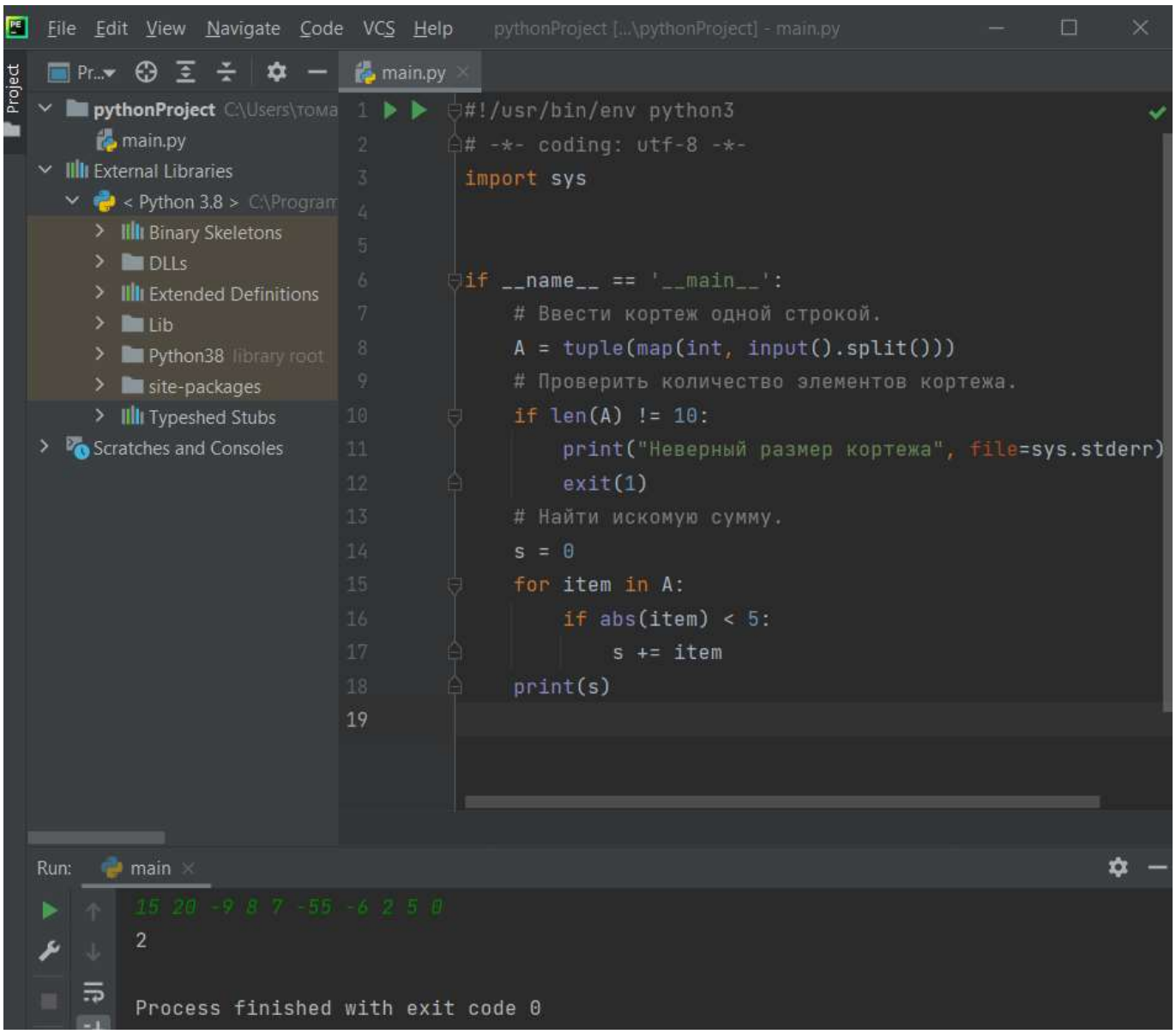

Рисунок 28 – Пример решения задачи: ввести кортеж A из 10 элементов, найти сумму элементов, меньших модулю 5, и вывести её на экран. Использовать в программе вместо списков кортежи.

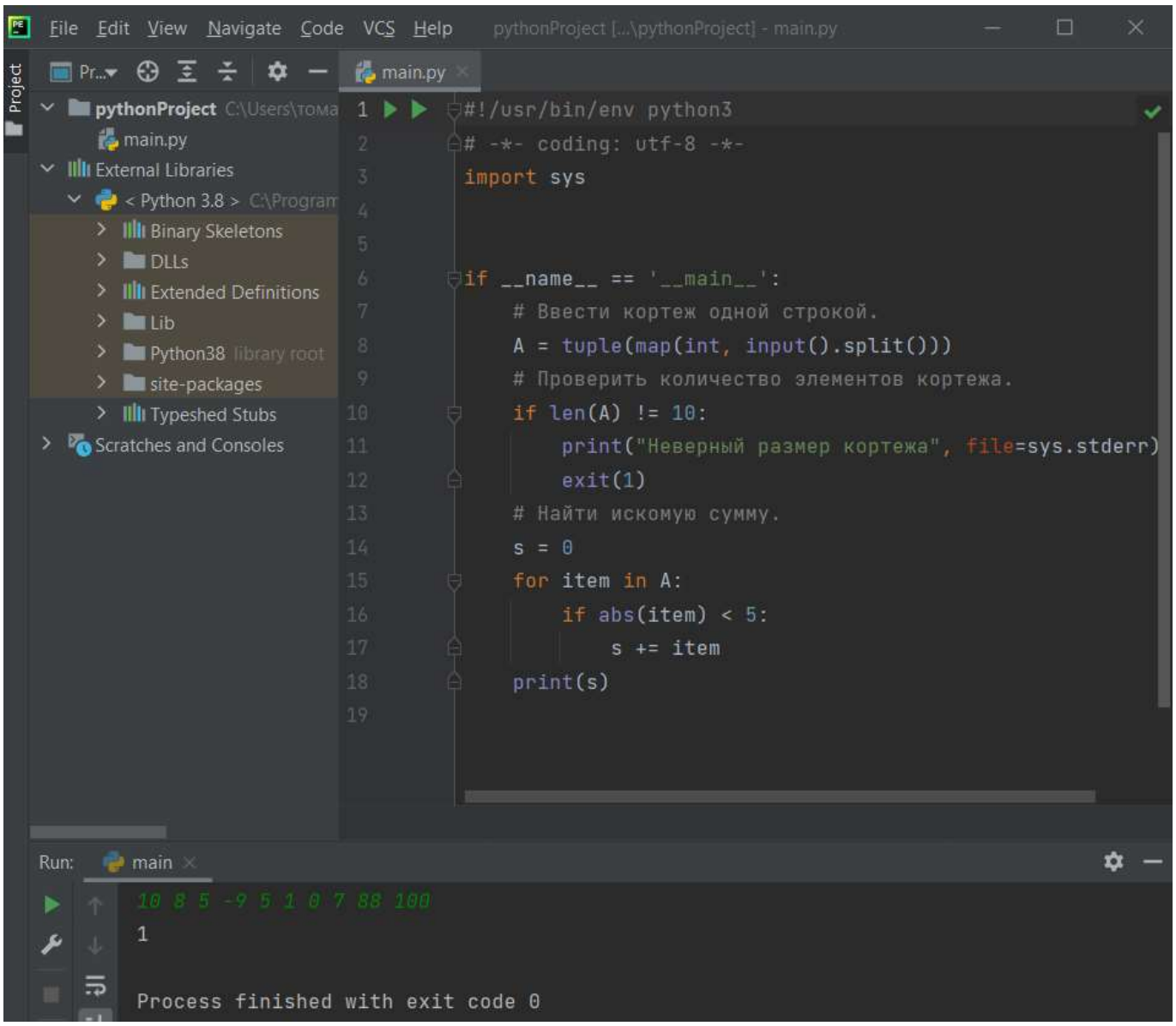

Рисунок 29 – Пример решения задачи: ввести кортеж A из 10 элементов, найти сумму элементов, меньших модулю 5, и вывести её на экран. Использовать в программе вместо списков кортежи.

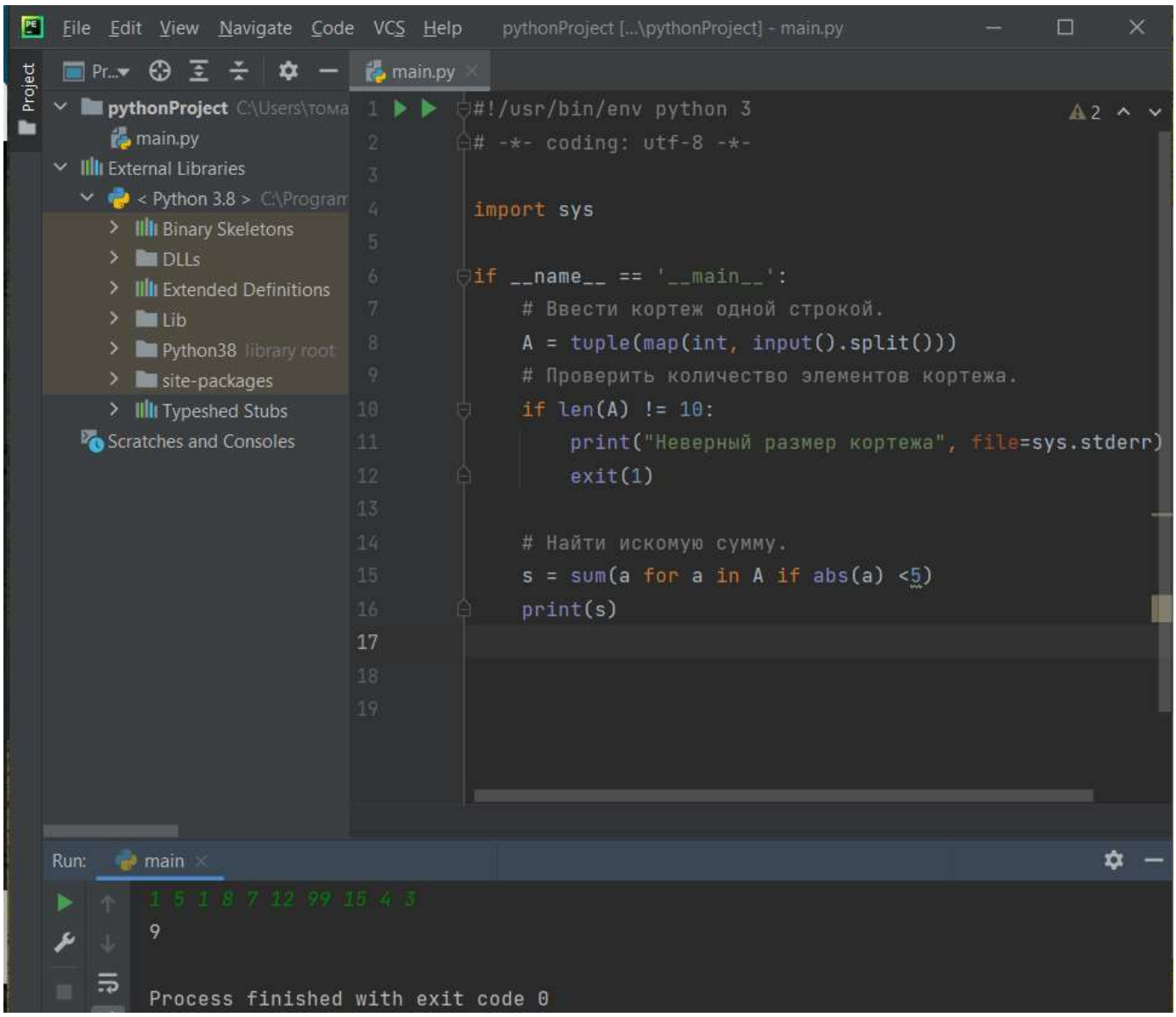

Рисунок 30 – Пример решения задачи с помощью списковых включений

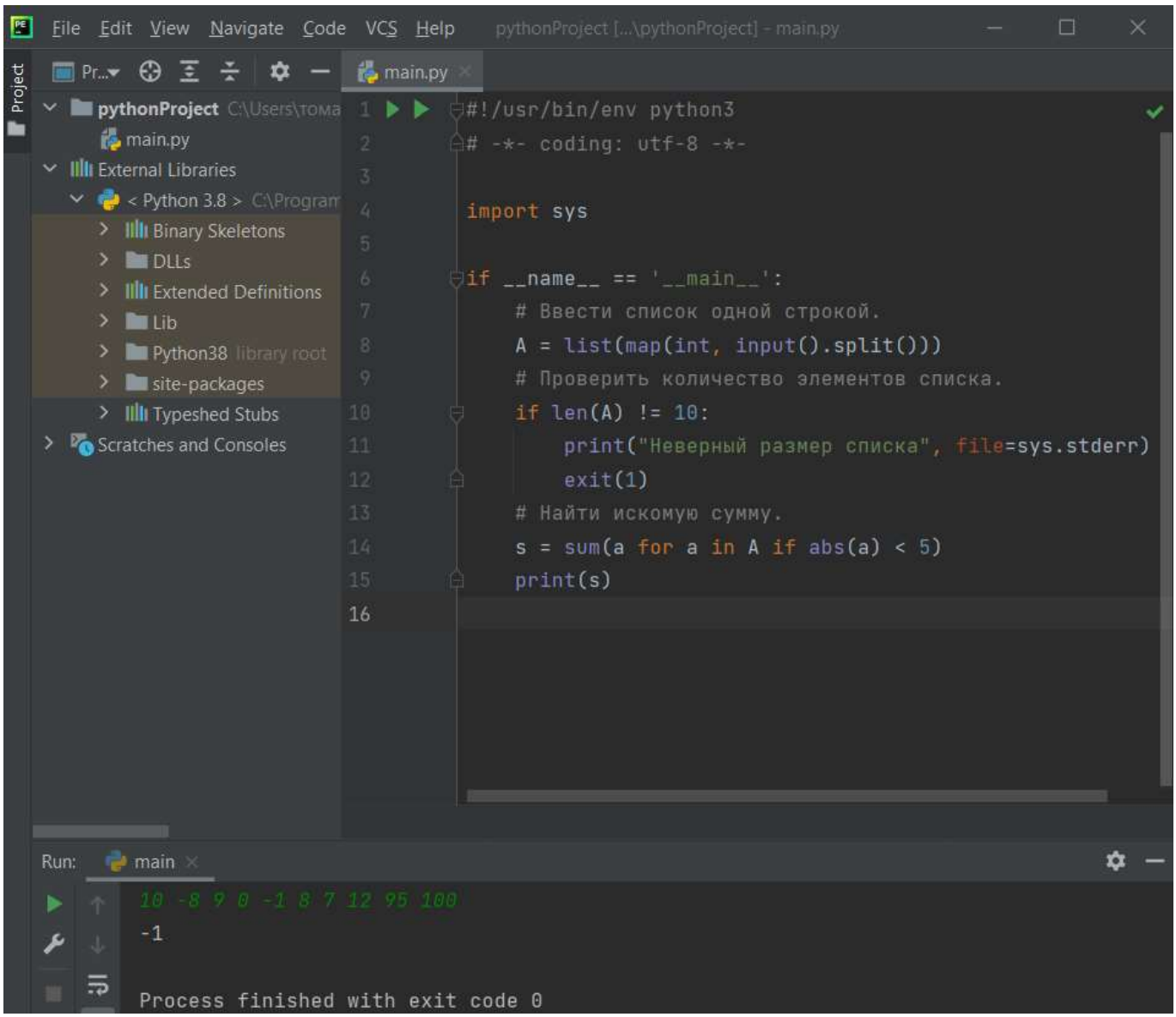

Рисунок 31 – Пример решения задачи с помощью списковых включений

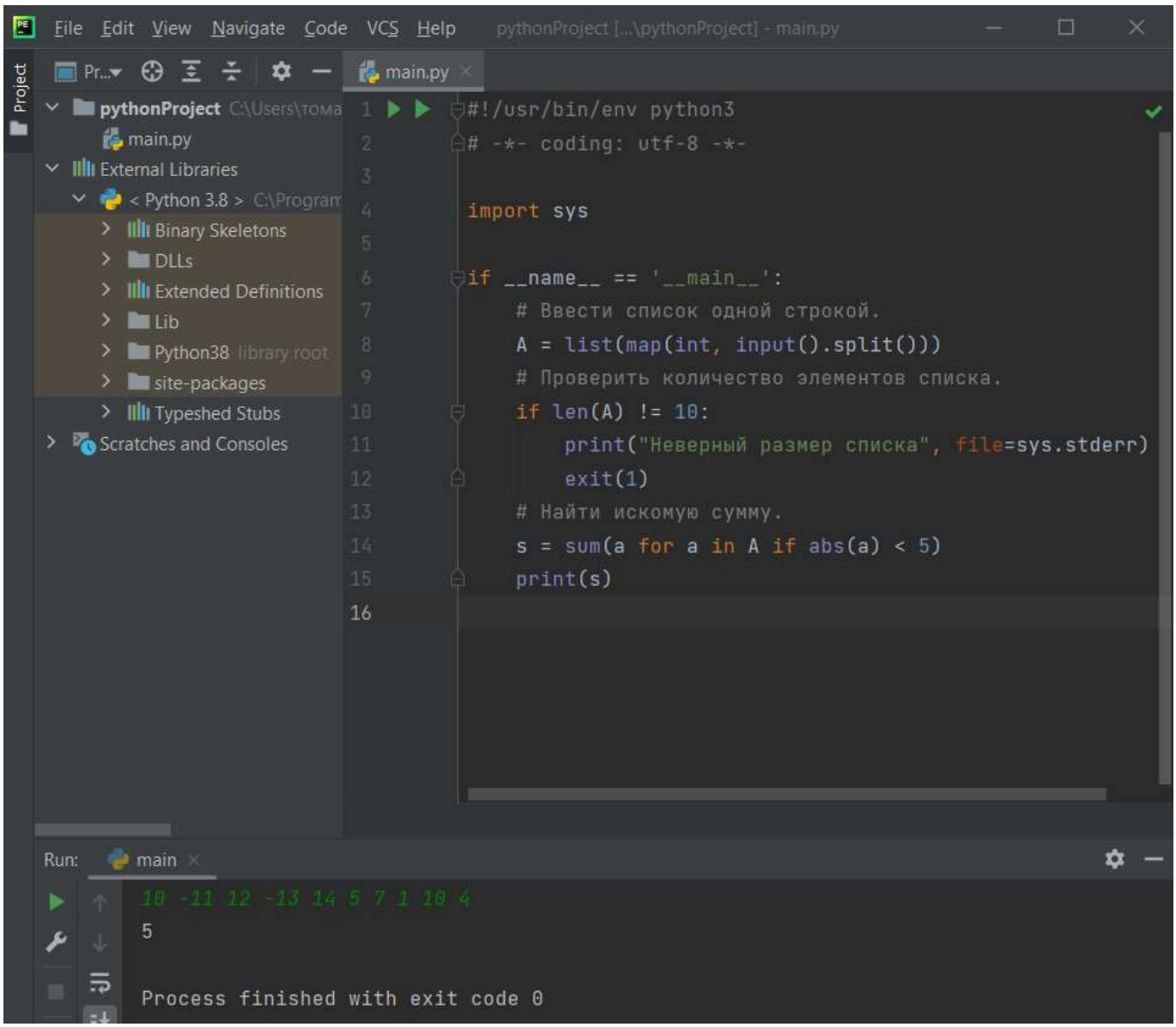

Рисунок 32 – Пример решения задачи с помощью списковых включений

3. Индивидуальное задание

Известно количество мячей, забитых футбольной командой за каждую игру в двух чемпионатах, которое хранится в двух кортежах. В каждом из чемпионатов команда сыграла 26 игр. Найти общее количество мячей, забитых командой в двух чемпионатах.

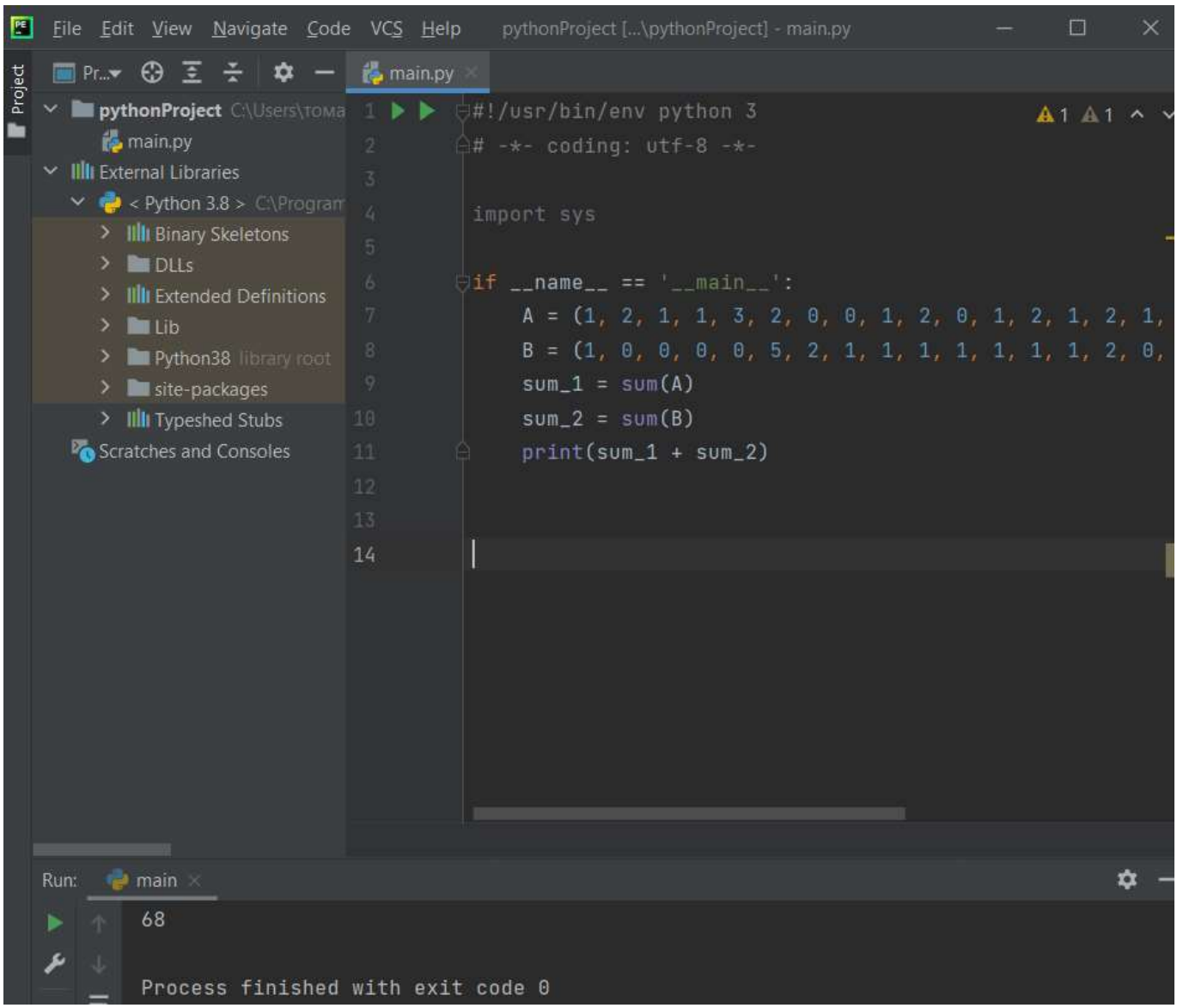

Рисунок 33 – Пример решения индивидуального задания

4. Вопросы для защиты работы

1. Что такое списки в языке Python?

Список – это изменяемый тип данных.

2. Каково назначение кортежей в языке Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похоже на список.

Назначение кортежей:

1) Обезопасить данные от случайного изменения.
2) Экономия места.
3) Прирост производительности, который связан с тем, что кортежи работают быстрее, чем списки.

3. Как осуществляется создание кортежей?

1) A = ()
2) A = tuple([1, 2 , 3])

4. **Как осуществляется доступ к элементам кортежа?**
   Доступ к элементам кортежа осуществляется через указание индекса. print(a[0]) или print(a[1:3])
5. **Зачем нужна распаковка (деструктуризация) кортежа?**
   Деструктуризация кортежа нужно для быстро разбиения кортежа на отдельные элементы, для более удобного доступа и работы.
6. **Какую роль играют кортежи в множественном присваивании?**
   Элементам кортежа можно сразу последовательно присваивать значения. А также используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.
7. **Как выбрать элементы кортежа с помощью среза?**
   Необходимо ввести: item = A[0:2], item = B[:3], item = B[1:]
   Общая форма операции взятия среза для кортежа:

```
T2 = T1[i:j]
```

8. **Как выполняется конкатенация и повторение кортежей?**
   Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +. Общая форма операции:

```
T3 = T1 + T2
```

   Кортеж может быть образован путём операции повторения, обозначаемой символом *. Общая форма операции:

```
T2 = T1 * n
```

9. **Как выполняется обход элементов кортежа?**
   Элементы кортежа можно последовательно просмотреть с помощью операторов цикла while или for. Пример представлен на рисунке 22.

```python
# Обход кортежа в цикле
# 1. Цикл for
# Заданный кортеж
A = ("abc", "abcd", "bcd", "cde")

# Вывести все элементы кортежа
for item in A:
    print(item)

# 2. Цикл while
# Исходный кортеж - целые числа
A = (-1, 3, -8, 12, -20)

# Вычислить количество положительных чисел
i = 0
k = 0 # количество положительных чисел

while i < len(A):
    if (A[i]<0):
        k = k +1
    i = i +1

# Вывести результат
print("k = ", k)
# 3. Обход в цикле for
# Заданный кортеж, содержащий строки
A = ("abc", "ad", "bcd")

# Сформировать новый список из элементов кортежа A
# в новом списке B, каждый элемент удваивается
B = [item * 2 for item in A]
```

10. Как проверить принадлежность элемента кортежу?

Проверить принадлежность элемента кортежу можно с помощью операции in.

```python
if (item in A):
    print(item, " in ", A, " = True")
else:
    print(item, " in ", A, " = False")
```

11. Какие методы работы с кортежами Вам известны?

Метод index(): pos = A.index(item)

Метод count(): k = A.count(item)

где item – элемент кортежа.

12. **Допустимо ли использование функций агрегации таких как len() sum() и т. д. при работе с кортежами?**
    Да, допустимо.
13. **Как создать кортеж с помощью спискового включения.**
    x = 10
    a = tuple([i for i in range(x)])

    a = tuple(int(i) for i in input().split())

Ссылка на репозиторий: https://github.com/tamaranesterenko/Python.LR_8