

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

**Отчет по лабораторной работе №4
Работа со строками в языке Python**

Выполнил студент группы

ИТС-б-з-22-1

Рябов З.А. « » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 2023г.

Проверил доцент, кандидат технических
наук, доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

(подпись)

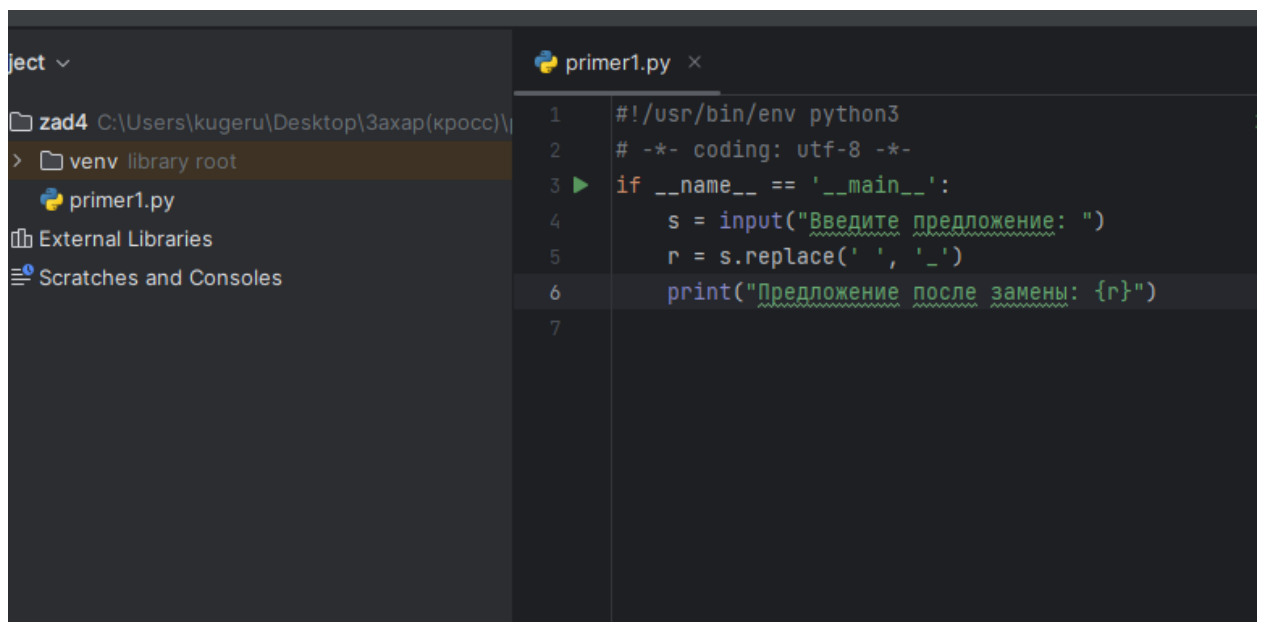
Ставрополь, 2023 г.

Цель работы: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python.

Порядок выполнения работы:

Создала общедоступный репозиторий на GitHub (<https://github.com/DemoGood/zadanie4>)

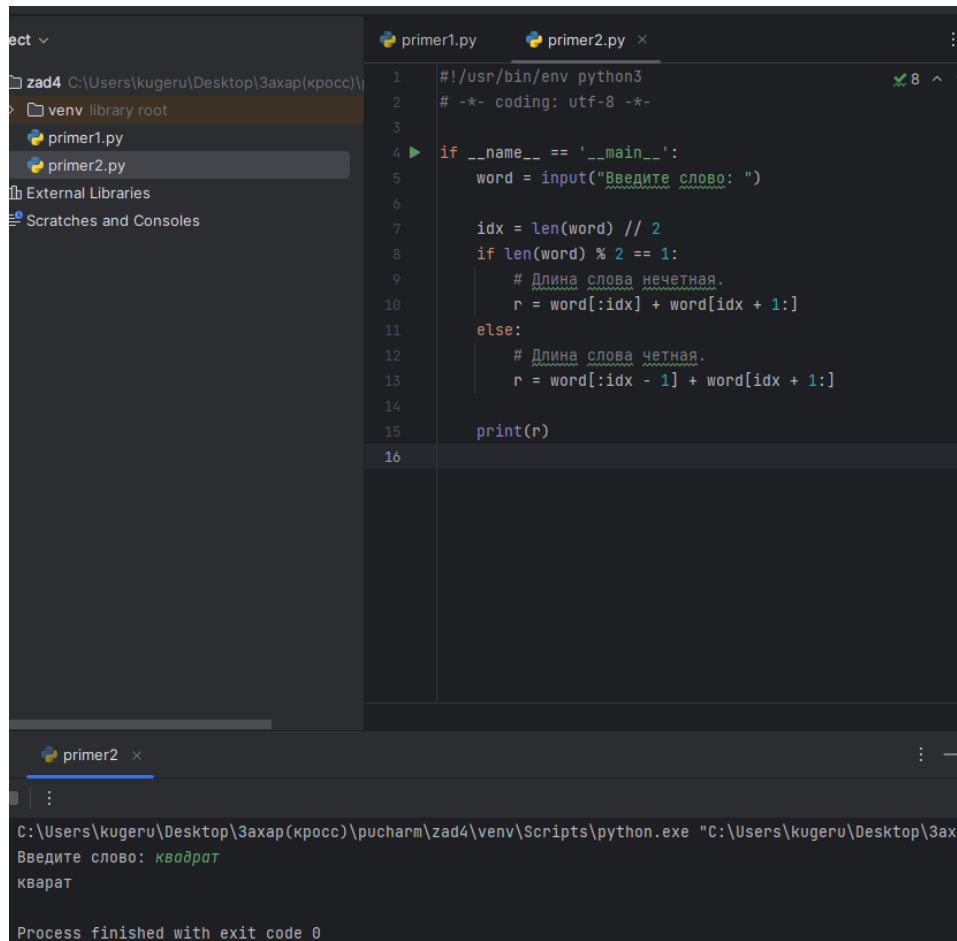
Пример 1. Дано предложение. Все пробелы в нем заменить символом «_».



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == '__main__':
4      s = input("Введите предложение: ")
5      r = s.replace(' ', '_')
6      print("Предложение после замены: {r}")
7
```

Рисунок 1 – Окно программы примера 1

Пример 2. Дано слово. Если его длина нечетная, то удалить среднюю букву, в противном случае – две средние буквы.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      word = input("Введите слово: ")
6
7      idx = len(word) // 2
8      if len(word) % 2 == 1:
9          # Длина слова нечетная.
10         r = word[:idx] + word[idx + 1:]
11     else:
12         # Длина слова четная.
13         r = word[:idx - 1] + word[idx + 1:]
14
15     print(r)
16
```

primer2

C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad4\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad4\venv\Scripts\python.exe"

Введите слово: *кварат*

кварат

Process finished with exit code 0

Рисунок 2 – Окно программы примера 2

Пример 3. Дана строка текста, в котором нет начальных и конечных пробелов. Необходимо изменить ее так, чтобы длина строки стала равна заданной длине (предполагается, что требуемая длина не меньше исходной). Это следует сделать путем вставки между словами дополнительных пробелов. Количество пробелов между отдельными словами должно отличаться не более чем на 1.

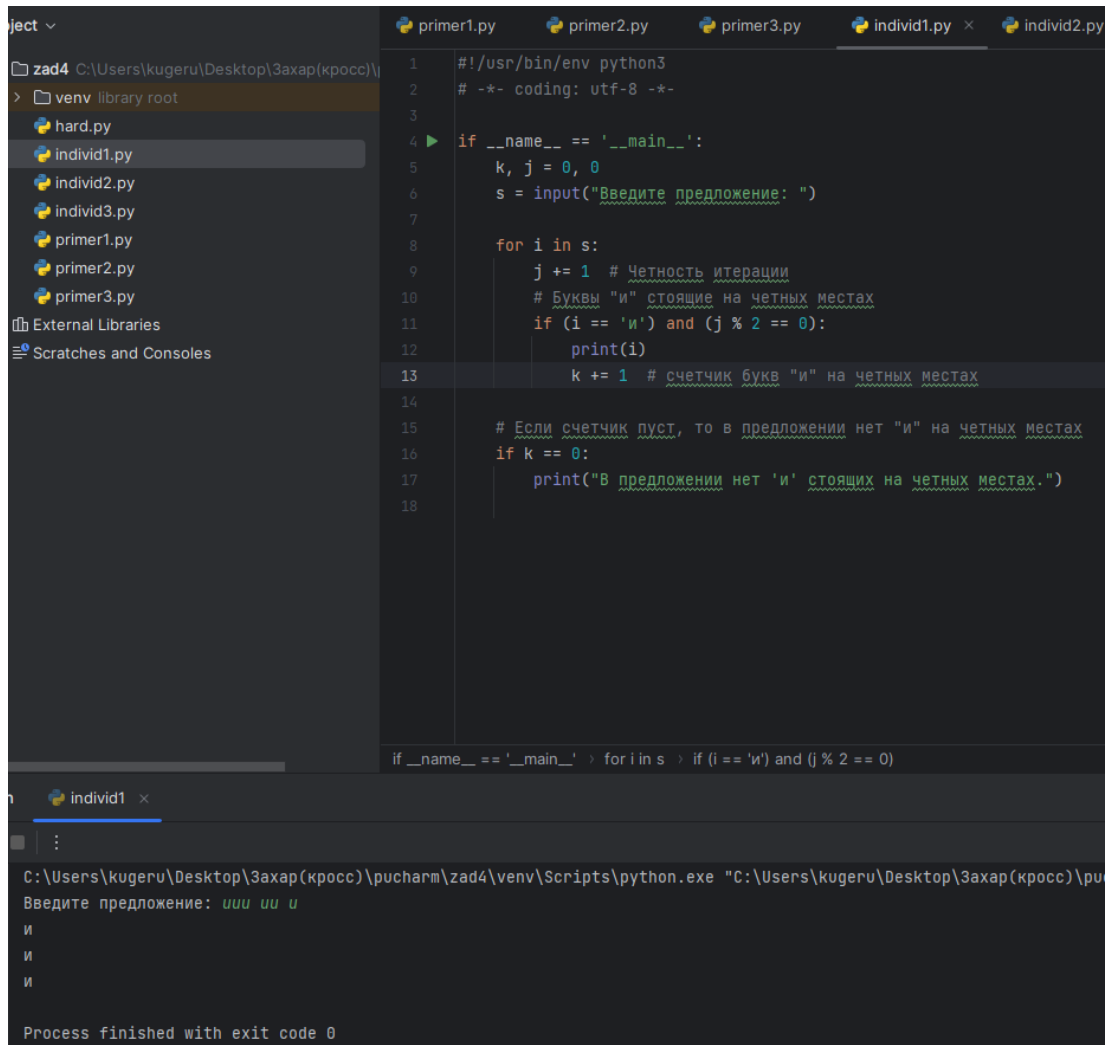
```
primer1.py  primer2.py  primer3.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      s = input("Введите предложение: ")
8      n = int(input("Введите длину: "))
9
10     # Проверить требуемую длину.
11     if len(s) >= n:
12         print(
13             "Заданная длина должна быть больше длины предложения",
14             file=sys.stderr
15         )
16         exit(1)
17
18     # Разделить предложение на слова.
19     words = s.split(' ')
20     # Проверить количество слов в предложении.
21     if len(words) < 2:
22         print(
23             "Предложение должно содержать несколько слов",
24             file=sys.stderr
25         )
26         exit(1)
27
28     # Количество пробелов для добавления.
29     delta = n
30     for word in words:
31         delta -= len(word)
32
33     # Количество пробелов на каждое слово.
34     w, r = delta // (len(words) - 1), delta % (len(words) - 1)
35
36     # Сформировать список для хранения слов и пробелов.
37     lst = []
38
39     # Пронумеровать все слова в списке и перебрать их.
40     for i, word in enumerate(words):
41         lst.append(word)
42
43         # Если слово не является последним, добавить пробелы.
44         if i < len(words) - 1:
45             # Определить количество пробелов.
46             width = w
47             if r > 0:
48                 width += 1
49                 r -= 1
50
51             # Добавить заданное количество пробелов в список.
52             if width > 0:
53                 lst.append(' ' * width)
54
55     # Вывести новое предложение, объединив все элементы списка lst.
56     print(''.join(lst))
57
```

```
:
Введите предложение: мама мыла раму
Введите длину: 20
мама    мыла    раму
> primer3.py
```

Рисунок 3 – Окно программы примера 3

Индивидуальное задание 1.

13. Дано предложение. Вывести «столбиком» все его буквы и, стоящие на четных местах.



The screenshot shows a Python IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'zad4' with a 'venv' directory and several Python files. The code editor displays a script named 'individ1.py' with the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     k, j = 0, 0
6     s = input("Введите предложение: ")
7
8     for i in s:
9         j += 1 # Четность итерации
10        # Буквы "и" стоящие на четных местах
11        if (i == 'и') and (j % 2 == 0):
12            print(i)
13        k += 1 # счетчик букв "и" на четных местах
14
15 # Если счетчик пуст, то в предложении нет "и" на четных местах
16 if k == 0:
17     print("В предложении нет 'и' стоящих на четных местах.")
18
```

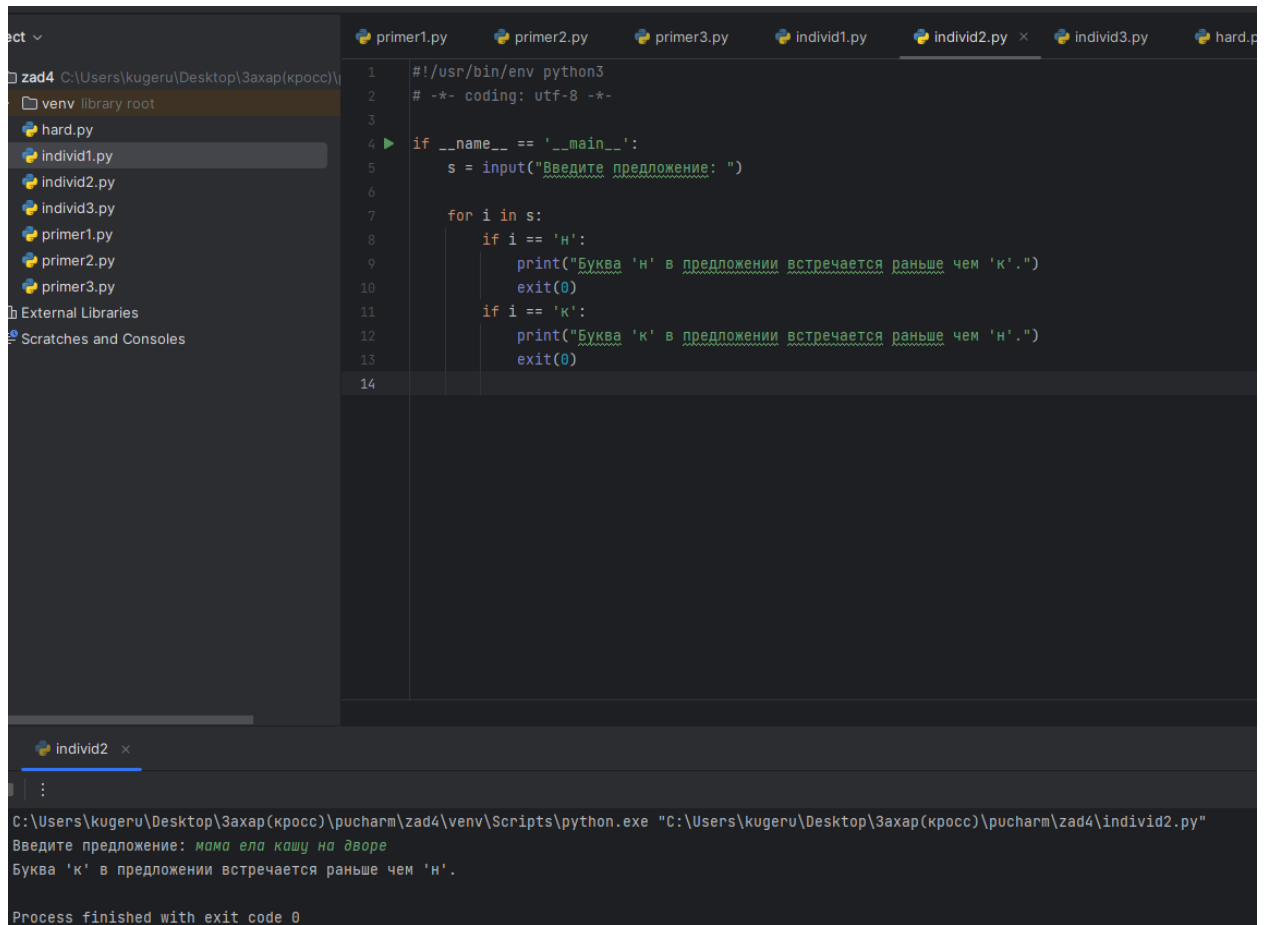
The console output shows the program running and the input 'uuu uu u' being processed, resulting in the output 'и' printed vertically.

```
C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad4\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Захар(кросс)\pu
Введите предложение: uuu uu u
и
и
и
Process finished with exit code 0
```

Рисунок 4 – Окно программы для первой задачи и проверка работоспособности кода.

Индивидуальное задание 2.

13. Дано предложение. Определить, какая из букв – н или к – встречается в ней раньше при просмотре слева направо (принять, что указанные буквы в строке есть).



The screenshot shows a Python IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'zad4' with a 'venv' directory and several Python files. The code editor shows the following Python code:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = input("Введите предложение: ")
6
7      for i in s:
8          if i == 'н':
9              print("Буква 'н' в предложении встречается раньше чем 'к'.")
10             exit(0)
11          if i == 'к':
12              print("Буква 'к' в предложении встречается раньше чем 'н'.")
13              exit(0)
14
```

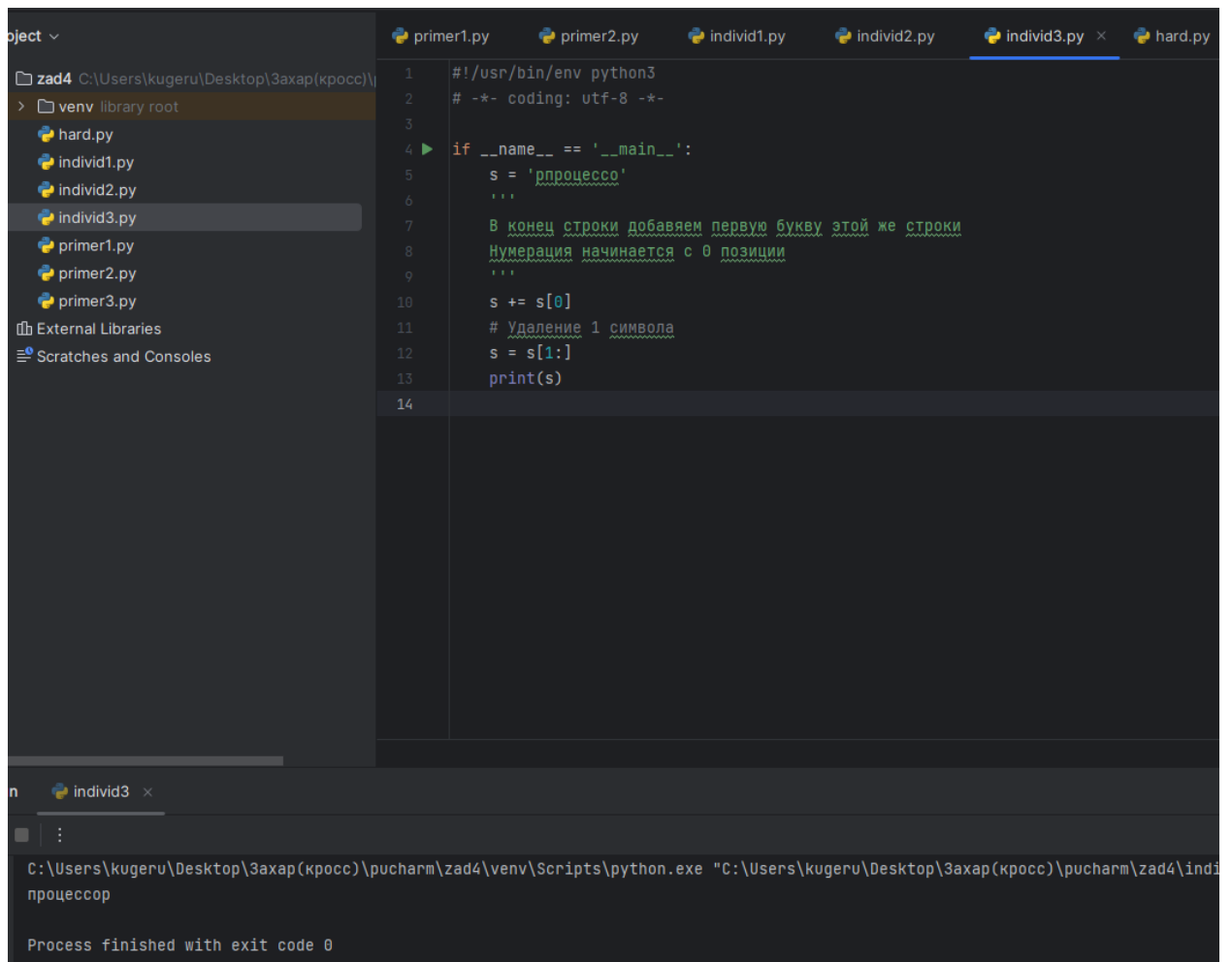
Below the code editor, there is a console window showing the execution of the script. The console output is as follows:

```
C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad4\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad4\individ2.py"
Введите предложение: мама ела кашу на дворе
Буква 'к' в предложении встречается раньше чем 'н'.
Process finished with exit code 0
```

Рисунок 5 – Окно программы для второй задачи и проверка работоспособности кода.

Индивидуальное задание 3.

13. Дано ошибочно написанное слово рпроцессо. Путем перемещения его букв получить слово процессор.



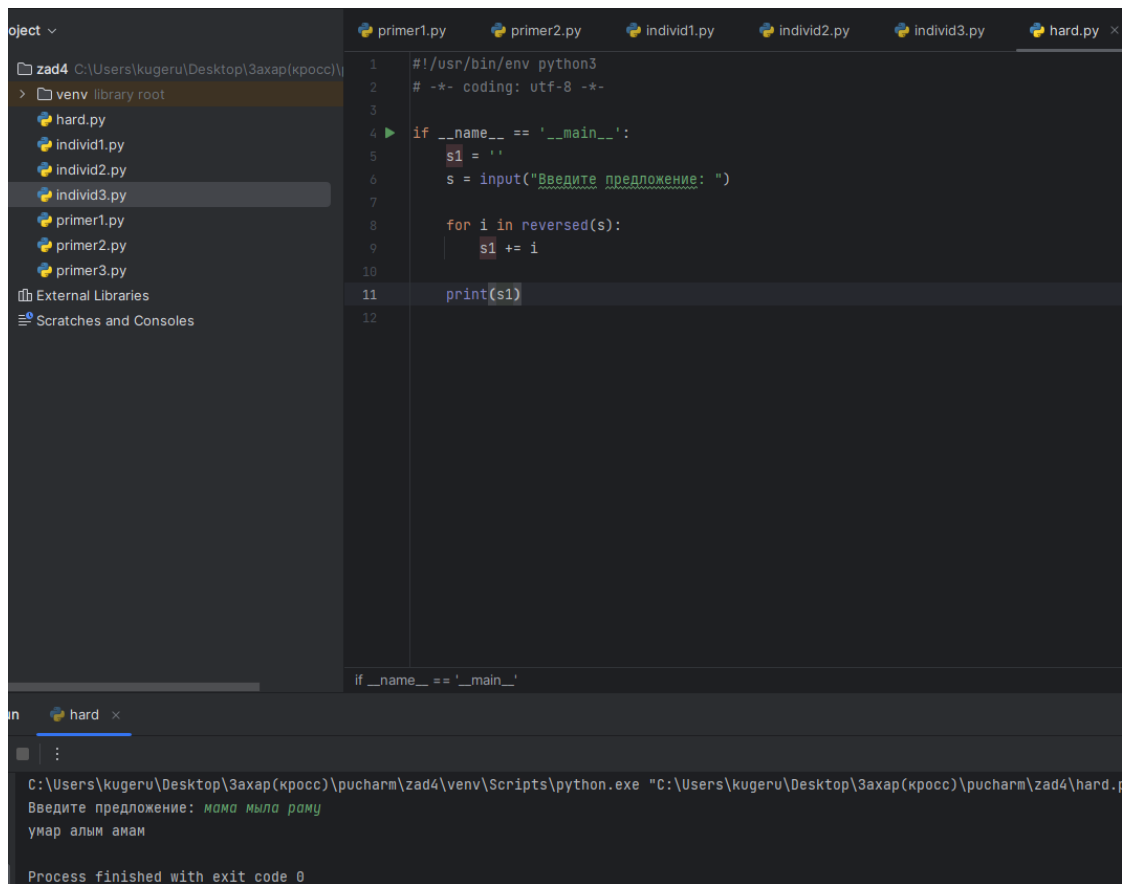
```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = 'рпроцессо'
6      '''
7      В конец строки добавляем первую букву этой же строки
8      Нумерация начинается с 0 позиции
9      '''
10     s += s[0]
11     # Удаление 1 символа
12     s = s[1:]
13     print(s)
14
```

Process finished with exit code 0

Рисунок 6 – Окно программы для третьей задачи и проверка работоспособности кода.

Задание повышенной сложности.

13. Дано предложение. Напечатать его в обратном порядке слов, например предложение мама мыла раму должно быть напечатано в виде раму мыла мама.



The screenshot shows a Python IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'zad4' with a 'venv' directory and several Python files. The code editor shows the following Python code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     s1 = ''
6     s = input("Введите предложение: ")
7
8     for i in reversed(s):
9         s1 += i
10
11 print(s1)
```

The console output shows the program running and the input 'мама мыла раму' being reversed to 'уар алым амам'.

Рисунок 7 – Окно программы для задания повышенной сложности и проверка работоспособности кода.

Вывод: приобрел навыки по работе со строками при написании программ с помощью языка программирования Python.