

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина: Избранные главы информатики

**ОТЧЁТ**  
к лабораторной работе 3  
на тему

Стандартные типы данных, коллекции, функции, модули в Python

Выполнил: студент группы 253501  
Борисевич Александр Михайлович

Проверила: Жвакина Анна Васильевна

Минск 2024

## **Оглавление**

<b>Оглавление .....</b>	<b>2</b>
<b>Цель работы .....</b>	<b>3</b>
<b>Ход работы.....</b>	<b>4</b>
<b>Выводы.....</b>	<b>16</b>

## **Вариант 4**

### **Цель работы**

освоить базовый синтаксис языка Python, приобрести навыки работы со стандартными типами данных, коллекциями, функциями, модулями и закрепить их на примере разработки интерактивных приложений.

## Ход работы

1. В соответствии с заданием своего варианта составить программу для вычисления значения функции с помощью разложения функции в степенной ряд. Задать точность вычислений  $\epsilon$ . Предусмотреть максимальное количество итераций, равное 500. Вывести количество членов ряда, необходимых для достижения указанной точности вычислений. Результат получить в виде:

$x$	$n$	$F(x)$	$\text{Math } F(x)$	$\epsilon$

Здесь  $x$  – значение аргумента,  $F(x)$  – значение функции,  $n$  – количество просуммированных членов ряда,  $\text{Math } F(x)$  – значение функции, вычисленное с помощью модуля `math`.

$$\ln(1-x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{n+1}}{n+1} = -x - \frac{x^2}{2} - \frac{x^3}{3} + \dots, |x| < 1$$

Модуль управления ходом выполнения задания:

```
56 def execute():
57     """
58     Execute the first task. Print results to console.
59
60     """
61
62     print("Task 1.")
63     print("Evaluate ln(1 - x).")
64
65     x = inps.input_specified_type(
66         | float, "Enter value of 'x': ")
67     eps = inps.input_specified_type(
68         | float, "Enter value of 'eps': ")
69
70     try:
71         | evaluate(x, eps)
72     except ValueError as e:
73         | print(e)
```

В модуле вычисления суммы ряда функции:

```

17 def evaluate(x, eps):
18     """
19     Prints a table with the values of x, n, F(x), the mathematical function F(x), and the desired accuracy eps.
20
21     Args:
22     |   x (float): Argument of the function.
23     |   eps (float): The desired accuracy.
24
25     Raises:
26     |   ValueError: If the absolute value of x is greater or equal to 1.
27     |   ValueError: If the value of eps is less or equal to 0.
28     """
29     print("Task 1.")
30     print("Evaluate ln(1 - x), |x| < 1")
31
32     if abs(x) >= 1:
33         raise ValueError("Absolute value of 'x' must be less than 1")
34     if eps <= 0:
35         raise ValueError("The value of 'eps' must be greater than 0")
36
37     print(f"x          n          F(x)          Math F(x)          eps")
38
39     n = 1
40     step = -x
41     result = -x
42     math_result = f(x)
43
44     for n in range(1, 501):
45         # Calculates the next value of x using the Newton-Raphson method
46         if abs(math_result - result) <= eps:
47             return
48
49         step *= x * n
50         step /= n + 1
51         result += step
52         # Prints the values of x, n, F(x), Math F(x), and eps
53         print(f"{'{:5f}'.format(x)}          {n}          {'{:5f}'.format(result)}          {'{:5f}'.format(math_result)}")
54

```

Вывод задания:

```

Task 1.
Evaluate ln(1 - x).
Enter value of 'x': 0.1928
Enter value of 'eps': 0.00001
Task 1.
Evaluate ln(1 - x), |x| < 1
x          n          F(x)          Math F(x)          eps
0.19280    1          -0.21139        -0.21418        0.00001000
0.19280    2          -0.21377        -0.21418        0.00001000
0.19280    3          -0.21412        -0.21418        0.00001000
0.19280    4          -0.21417        -0.21418        0.00001000
0.19280    5          -0.21418        -0.21418        0.00001000
Do you want to continue this task: (y/n): S

```

2. В соответствии с заданием своего варианта составить программу для нахождения суммы последовательности чисел.

- |    |   |
|----|---|
| 4. | Организовать цикл, который принимает целые числа с клавиатуры и подсчитывает количество четных чисел. Окончание цикла – ввод числа, большего 1000 |
|----|---|

Модуль управления ходом выполнения задания:

```
36 def execute():
37     """
38     Execute the second task. Print results to console.
39     """
40
41     print("Task 2.")
42     print("Calculating sum of list and finding amount of even numbers in a list.")
43     print("Enter a number greater or equal to 1000 to stop adding numbers.")
44     print()
45
46     print("Do you want to generate sequence automatically? (y/n): ")
47     choice = input().lower()
48
49     if choice == "y" or choice == "yes":
50         sequence_size = inp.input_specified_type(
51             | int, "Enter amount of numbers in a sequence: ")
52
53         numbers = list(inpseq.sequence_generator(sequence_size))
54     else:
55         numbers = inpseq.input_by_maximum_number(max_value_of_number)
56
57     print(f"Your numbers: { numbers }")
58     print(f"Total sum of this list: { sum_of_numbers(numbers) }")
59     print(f"This list contains { amount_of_even_numbers(numbers) } even numbers")
```

Модуль инициализации последовательности состоит из двух функций:

```
28 def input_by_maximum_number(max_value_of_number: int):
29     """
30     Returns a list of numbers entered by the user.
31     If the user enters a number greater than the maximum number, input ends.
32
33     Args:
34     |     max_value_of_number (int): The maximum number.
35
36     Returns:
37     |     list[int]: The list of numbers entered by the user.
38
39     """
40
41     numbers = []
42
43     while True:
44         number = inp.input_specified_type(int, "Enter a number: ")
45
46         # Check if the number is greater than maximum.
47         if number > max_value_of_number:
48             break
49
50         numbers.append(number)
51
52     return numbers
53
54 def sequence_generator(last_number: int):
55     """
56     Returns a generator that generates a sequence of numbers.
57
58     Args:
59     |     last_number (int): The last number in the sequence.
60
61     Returns:
62     |     generator[int]: The generator that generates a sequence of numbers.
63
64     """
65     for number in range(1, last_number + 1):
66         yield number
```

Модуль функций для выполнения задания:

```
4 # Maximum value for the sequence.
5 max_value_of_number = 1000
6
7
8 def sum_of_numbers(numbers: list[int]) -> int:
9     """
10     Finds the sum of a list of numbers.
11
12     Args:
13     |     numbers (list[int]): The list of numbers.
14
15     Returns:
16     |     int: The sum of the numbers in the list.
17     """
18
19     return sum(numbers)
20
21
22 def amount_of_even_numbers(numbers: list[int]) -> int:
23     """
24     Finds the amount of even numbers in a list.
25
26     Args:
27     |     numbers (list[int]): The list of numbers.
28
29     Returns:
30     |     int: The amount of even numbers in the list.
31     """
32
33     return sum(map(lambda num: 1 if num % 2 == 0 else 0, numbers))
```

Вывод задания:

```
Task 2.
Calculating sum of list and finding amount of even numbers in a list.
Enter a number greater or equal to 1000 to stop adding numbers.

Do you want to generate sequence automatically? (y/n):
n
Enter a number: 12
Enter a number: 13
Enter a number: 14
Enter a number: 9999999999
Your numbers: [12, 13, 14]
Total sum of this list: 39
This list contains 2 even numbers
Do you want to continue this task: (y/n):
```

3. Не использовать регулярные выражения. В соответствии с заданием своего варианта составить программу для анализа текста, вводимого с клавиатуры.



- |    |  |
|----|--|
| 4. | В строке, вводимой с клавиатуры, подсчитать количество пробельных символов |
|----|--|

Модуль управления ходом выполнения задания:

```
30 def execute():
31     """
32     Execute the third task. Print results to console.
33     """
34
35     print("Task 3.")
36     print("Count spaces in a string.")
37     print()
38
39     string = input("Enter a string: ")
40
41     print(f"Your string: '{string}'")
42     output = output_decorator(amount_of_spaces)
43     print(output(string))
44
```

Демонстрационный модуль декоратора:

```
14 def output_decorator(function):
15     """
16     Decorates a function that outputs a string.
17
18     Args:
19     |   function (function): The function to decorate.
20
21     Returns:
22     |   function: The decorated function.
23     """
24
25     def wrapper(s):
26         return " ".join(["The string contains", str(function(s)), "spaces."])
27
28     return wrapper
```

Модуль для подсчёта пробелов в строке:

```
1  def amount_of_spaces(string: str) -> int:
2      """
3          Counts the number of spaces in a string.
4
5          Args:
6          |   string (str): The string.
7
8          Returns:
9          |   int: The number of spaces in the string.
10     """
11
12     return len(string.split(' ')) - 1
```

Вывод задания:

```
Task 3.
Count spaces in a string.

Enter a string: aboba  aboba aboba
Your string: 'aboba  aboba aboba'
The string contains 3 spaces.
Do you want to continue this task: (y/n): ☐
```

4. **Не использовать регулярные выражения.** Дана строка текста, в которой слова разделены пробелами и запятыми. В соответствии с заданием своего варианта составьте программу для анализа строки, инициализированной в коде программы: «So she was considering in her own mind, as well as she could, for the hot day made her feel very sleepy and stupid, whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.» Если не оговорено иное, то регистр букв при решении задачи не имеет значения.

4.	а) определить количество слов, длина которых меньше 5; б) найти самое короткое слово, заканчивающееся на букву 'd'; в) вывести все слова в порядке убывания их длин
----	---

Модуль управления ходом выполнения задания:

```

46 def execute():
47     """
48     Executes the fourth task. Prints results to console.
49     """
50
51     print("Task 4.")
52     print("Count words with less than 5 letters.")
53     print("Prints shortest word that ends with letter 'd'.")
54     print("Prints words in descending order.")
55
56     text = ('So she was considering in her own mind, as well as she could, for the hot day made her feel '
57            | 'very sleepy and stupid, whether the pleasure of making a daisy-chain would be worth the trouble '
58            | 'of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by '
59            | 'her.')
60
61     print(('Amount of words with less than 5 letters is'
62           | f' { amount_of_words(text, lambda word: len(word) < 5) }.'))
63     print()
64
65     print(('Shortest word that ends with letter '
66           | f'"d" - "{ shortest_word(text, lambda word: word[-1].lower() == "d") }".'))
67     print()
68
69     print("Words by length in decreasing order: ")
70     print("-- ", end='')
71     print('\n-- '.join(words_by_length_in_decreasing_order(text)))
72

```

Модуль для анализа текста содержит три функции:

```

1  def amount_of_words(text: str, condition) -> int:
2      """
3      Counts the number of words in a string that satisfy a condition.
4
5      Args:
6      |   text (str): The string.
7      |   condition (function): The condition.
8
9      Returns:
10     |   int: The number of words in the string that satisfy the condition.
11     """
12
13     return sum([1 if word.isalpha() and condition(word) else 0 for word in text.split(" ")])
14
15
16 def shortest_word(text: str, condition) -> str:
17     """
18     Finds the shortest word in a string that satisfies a condition.
19
20     Args:
21     |   text (str): The string.
22     |   condition (function): The condition.
23
24     Returns:
25     |   str: The shortest word in the string that satisfies the condition.
26     """
27
28     return min([word for word in text.split(" ") if word.isalpha() and condition(word)])
29

```

```

31 def words_by_length_in_decreasing_order(text: str) -> list[str]:
32     """
33     Returns the words in a string in decreasing order of their length.
34
35     Args:
36     |     text (str): The string.
37
38     Returns:
39     |     list[str]: The words in the string in decreasing order of their length.
40     """
41
42     return sorted([word for word in text.split(" ") if word.isalpha()],
43                   | | | | key=len, reverse=True)
44

```

Вывод задания:

```

Task 4.
Count words with less than 5 letters.
Prints shortest word that ends with letter 'd'.
Prints words in descending order.
Amount of words with less than 5 letters is 35.

Shortest word that ends with letter "d" - "and".

Words by length in decreasing order:
-- considering
-- pleasure
-- suddenly
-- whether
-- trouble
-- getting
-- picking
-- sleepy
-- making
-- Rabbit
-- would
-- worth
-- White
-- close
-- well

```

5. В соответствии с заданием своего варианта составить программу для обработки вещественных списков. Программа должна содержать следующие базовые функции:

- 1) ввод элементов списка пользователем;
- 2) проверка корректности вводимых данных;
- 3) реализация основного задания с выводом результатов;
- 4) вывод списка на экран.

4. Найти номер максимального по модулю элемента списка и произведение элементов, расположенных между первым и вторым нулевыми элементами

Модуль управления ходом выполнения задания:

```
45 def execute():
46     """
47     Execute the fifth task. Prints results to console.
48     """
49     print("Task 5.")
50     print("Finds index of maximum number.")
51     print("Finds multiplication of numbers between zeroes.")
52
53     numbers = inpseq.default_input(float)
54
55     print(f"Your numbers: { numbers }")
56     print()
57
58     print(f"Index of maximum is { find_index_of_maximum(numbers) }")
59     try:
60         print(f"Multiplication of elements between zeroes is { multiplication_of_elements_between_zeroes(numbers) }.")
61     except ValueError as e:
62         print(e)
63     print()
```

Модуль для работы со списком содержит четыре функции:

```
4 def find_index_of_maximum(nums: list[float]) -> int:
5     """
6     Finds the index of the maximum value in a list of numbers.
7
8     Args:
9     |   nums (list[float]): The list of numbers.
10
11     Returns:
12     |   int: The index of the maximum value in the list.
13     """
14
15     return 1 + nums.index(max(nums))
```

```

17 def multiplication_of_elements_between_zeroes(nums: list[float]) -> float:
18     """
19     Finds the multiplication of elements between zeroes in a list of numbers.
20
21     Args:
22     |     nums (list[float]): The list of numbers.
23
24     Raises:
25     |     ValueError: If the list does not contain any zeroes.
26     |     ValueError: If the list contains only one zero.
27
28     Returns:
29     |     float: The multiplication of elements between zeroes in the list.
30     """
31
32     if not 0.0 in nums:
33         raise ValueError("List does not contain any zeroes.")
34
35     if nums.index(0.0) == len(nums) - 1 - nums[::-1].index(0.0):
36         raise ValueError("List contains only one zero.")
37
38     # Reduce the list of numbers between first and last
39     # zero in input list to their multiplication.
40     return reduce(lambda x, y: x * y,
41                  [num for num in nums
42                   [nums.index(0.0) + 1 : len(nums) - 1 - nums[::-1].index(0.0)]],
43                  1.0)

```

```

3 def default_input(number_type: type):
4     """
5     Returns a list of numbers entered by the user.
6
7     Args:
8     |     number_type (type): The type of the numbers in the list.
9
10    Returns:
11    |     list[number_type]: The list of numbers entered by the user.
12    """
13
14    numbers = []
15
16    while True:
17        number = inp.input_specified_type(number_type, "Enter a number: ")
18        # Add entered number to the list.
19        numbers.append(number)
20
21        choose = input("Do you want to continue? (y/n): ").lower()
22        if choose != "y" and choose != "yes":
23            break
24
25    return numbers

```

Вывод задания:

```
Task 5.  
Finds index of maximum number.  
Finds multiplication of numbers between zeroes.  
Enter a number: 1.1  
Do you want to continue? (y/n): y  
Enter a number: 0.0  
Do you want to continue? (y/n): y  
Enter a number: 2.0  
Do you want to continue? (y/n): y  
Enter a number: 3.0  
Do you want to continue? (y/n): y  
Enter a number: 0.0  
Do you want to continue? (y/n): y  
Enter a number: 4.91  
Do you want to continue? (y/n): n  
Your numbers: [1.1, 0.0, 2.0, 3.0, 0.0, 4.91]  
  
Index of maximum is 6  
Multiplication of elements between zeroes is 6.0.
```

## **Выводы**

В ходе выполнения данной лабораторной работы был освоен базовый синтаксис языка Python, приобретены навыки работы со стандартными типами данных, коллекциями, функциями, модулями и закреплены их на примере разработки интерактивных приложений. Были выполнены задания, затрагивающие все вышеперечисленные аспекты программирования на языке Python.