Лабораторная работа №3. Стандартные типы данных, коллекции, функции, модули.
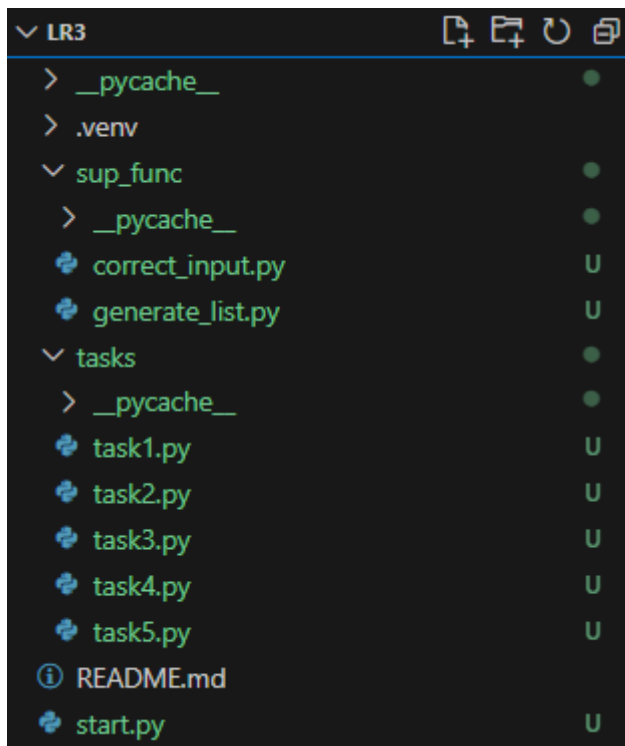
Дриневский Кирилл гр.253502. Репозиторий: https://github.com/Drinevskiy/IGI-STRWEB.git

Структура проекта:

Вспомогательные функции: correct_input.py, generate_list.py

```python
sup_func > 🐍 correct_input.py > 📦 input_float
1    def input_int():
2        """
3        Input int and check it.
4
5        Return:
6        number(int): entered number
7        """
8        number = 0
9        while(True):
10           try:
11               number = int(input())
12               break
13           except ValueError:
14               print("Incorrect input. Try again.")
15       return number
16
17   def input_float():
18       """
19       Input float and check it.
20
21       Return:
22       number(float): entered number
23       """
24       number = 0
25       while(True):
26           try:
27               number = float(input())
28               break
29           except ValueError:
30               print("Incorrect input. Try again.")
31       return number
```

```python
import random
import sup_func.correct_input

def generate_float_list(size):
    """
    Generate list of random float value.

    Parameters:
    size(int): length of list

    Return:
    float_list(list): generated list with random values
    """
    index = 0
    float_list = list()
    while(index < size):
        float_list.append(random.uniform(-100000,100000))
        index+=1
    return float_list

def input_float_list(size):
    """
    Manually filling in the list by the user.

    Parameters:
    size(int): length of list

    Return:
    float_list(list): filled list with user's values
    """
    a = 1
    index = 0
    float_list = list()
    while(index < size):
        print("Input value:", end = " ")
        a = sup_func.correct_input.input_float()
        float_list.append(a)
        index+=1
    return float_list
```

Меню: start.py

```python
# The program is designed to perform tasks from laboratory
# work №3 "Standard data types, collections, functions, modules.".
# Author: Drinevskiy Kirill
# Date: 27.03.2024
# Version: 1.0
from tasks.task1 import task1
from tasks.task2 import task2
from tasks.task3 import task3
from tasks.task4 import task4
from tasks.task5 import task5
from sup_func.correct_input import input_int

while(True):
    print("First task - 1\nSecond task - 2\nThird task - 3\nFourth task - 4\nFifth task - 5\nExit - 6")
    print("Choose number to perform task", end = " ")
    choose = input_int()
    match choose:
        case 1:
            task1()
        case 2:
            task2()
        case 3:
            task3()
        case 4:
            task4()
        case 5:
            task5()
        case 6:
            break
        case _:
            print("Incorrect input. Try again.")
print("Exit")
```

1. *Составить программу для вычисления значения функции arcsin с помощью разложения функции в степенной ряд. Задать точность вычислений eps.*

```python
tasks > 🐍 task1.py > 🔵 arcsin_series
  1   import math
  2   from sup_func.correct_input import input_float
  3
  4   def check_range(input_func):
  5       def output_func(*args):
  6           x = args[0]
  7           eps = args[1]
  8           if(abs(x) > 1):
  9               print("The value of x is out of bounds. x will be assigned the value of the nearest boundary.")
 10               if(x > 1):
 11                   x = 1
 12               else:
 13                   x = -1
 14           return input_func(x, eps)
 15       return output_func
 16
 17   @check_range
 18   def arcsin_series(x, eps):
 19       """
 20       Calculates the value of the arcsin using a series expansion with a given accuracy.
 21
 22       Parameters:
 23       x(float): argument of function arcsin
 24       eps(float): accuracy
 25
 26       Return:
 27       result(float): function value
 28       n(int): count of iterations
 29       """
 30       n = 0
 31       term = x
 32       result = x
 33       while abs(term) > eps and n < 500:
 34           n += 1
 35           term *= x * x * (2 * n - 1) * (2 * n - 1) / (2 * n * (2 * n + 1))
 36           result += term
 37       return result, n
 38
 39   def task1():
 40       """
 41       Perform first task.
 42       """
 43       print("Calculates the value of the arcsin using a series expansion with a given accuracy.")
 44       print("Input value x:", end = " ")
 45       x = input_float()
```

```python
 46       print("Input accuracy eps:", end = " ")
 47       eps = input_float()
 48       result, n = arcsin_series(x, eps)
 49       x1 = x
 50       print(f"Row: arcsin({x}) = {result}")
 51       try:
 52           print(f"Module math: arcsin({x1}) = {math.asin(x)}")
 53       except:
 54           if(x > 1):
 55               x = 1
 56           else:
 57               x = -1
 58           print(f"Module math: arcsin({x1}) = {math.asin(x)}")
 59       print(f"Number of row members = {n}")
 60       print(f"Accuracy = {eps}")
 61       print()
```

2. Организовать цикл, который принимает целые числа и суммирует каждое второе из них. Окончание цикла – ввод числа 0

```python
tasks > 🐍 task2.py > 📦 task2
  1   from sup_func.correct_input import input_int
  2
  3   def sum_second_number():
  4       """
  5       Sums up every second number.
  6       The end of the cycle is the input of the number 0.
  7
  8       Return:
  9       res(int): sum of every second number
 10       """
 11       a = 1
 12       even = False
 13       res = 0
 14       while(a!=0):
 15           print("Input value:", end = " ")
 16           a = input_int()
 17           if(even):
 18               res += a
 19           even = not even
 20       return res
 21
 22   def task2():
 23       """
 24       Perform second task.
 25       """
 26       print("Sums up every second number")
 27       print(f"Result: {sum_second_number()}")
```

3. *В строке, вводимой с клавиатуры, подсчитать количество цифр.*

```python
tasks >  task3.py > ...
 1    numbers = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'}
 2    def find_count_of_numbers_in_text(text):
 3        """
 4        Counting the number of digits in string.
 5
 6        Parameters:
 7        text(string): text
 8
 9        Return:
10        res(int): count of number in string
11        """
12        res = 0
13        for c in text:
14            if(c in numbers):
15                res+=1
16        return res
17
18    def task3():
19        """
20        Perform third task.
21        """
22        print("Counting the number of digits in string")
23        text = input("Input text: ")
24        res = find_count_of_numbers_in_text(text)
25
26        print(f"Count of numbers in text: {res}")
27        print()
```

4. *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.»
   а) определить число слов, длина которых равна 3 символа;
   б) найти слова, у которых количество гласных равно количеству согласных и их порядковые номера;
   в) вывести слова в порядке убывания их длин

```python
def delete_punctuation(list_of_string):
    """
    Delete "." and "," from each string of the list if they are present there.

    Parameter:
    list_of_string(list): list of strings

    Return:
    temp_list(list): list of strings without punctuation.
    """
    temp_list = []
    for item in list_of_string:
        if(item[-1]=="," or item[-1]=="."):
            item = item[:-1]
        temp_list.append(item)
    return temp_list

def count_of_words_with_length(list_of_string, length):
    """
    Counts the number of words with a given length.

    Parameters:
    list_of_string(list): list of strings
    length(int): length of word

    Return:
    res(int): count of words with given length
    """
    res = 0
    for item in list_of_string:
        if(len(item) == length):
            res+=1
    return res

def sort_text_by_words_length(list_of_string):
    """
    Sorts the list of strings in descending order of length.

    Parameters:
    list_of_string(list): list of strings

    Return:
    list_of_string(list): list of strings in descending order of length
    """
    return sorted(list_of_string, key=len, reverse=True)
```

```python
47  def dictionary_of_vowel_consonant_equal(list_of_string):
48      """
49      Returns a dictionary of words with the same number of consonants and vowels and their indexes.
50
51      Parameters:
52      list_of_string(list): list of strings
53
54      Return:
55      dictionary(dict): dictionary(key - word, value - index) of words
56      with the same number of consonants and vowels and their indexes.
57      """
58      length = len(list_of_string)
59      index = 0
60      vowels = "aeiou"
61      dictionary = dict()
62      while index < length:
63          word = list_of_string[index].lower()
64          vowel_count = len([letter for letter in word if letter in vowels])
65          consonant_count = len([letter for letter in word if letter not in vowels])
66          if(vowel_count == consonant_count):
67              dictionary[list_of_string[index]] = index + 1
68          index+=1
69      return dictionary
70
71  def task4():
72      """
73      Perform fourth task.
74      """
75      text = """So she was considering in her own mind, as well as she could, for the hot day made
76              her feel very sleepy and stupid, whether the pleasure of making a daisy-chain would be
77              worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit
78              with pink eyes ran close by her."""
79      split_text = text.split()
80      split_text = delete_punctuation(split_text)
81
82      count_of_words_with_length_3 = count_of_words_with_length(split_text, 3)
83      print(f"Count of words with length 3: {count_of_words_with_length_3}.")
84      print()
85      print("Dictionary(key - word, value - index) of words with the same number of consonants and vowels and their indexes.")
86      dictionary = dictionary_of_vowel_consonant_equal(split_text)
87      print(dictionary)
88      print()
89      print("List of strings in descending order of length.")
90      sorted_text = sort_text_by_words_length(split_text)
91      print(sorted_text)
```

5. *Найти количество положительных четных элементов списка и сумму элементов списка, расположенных после последнего элемента, равного нулю*

```python
1   from sup_func.generate_list import generate_float_list
2   from sup_func.generate_list import input_float_list
3   from sup_func.correct_input import input_int
4
5   def sum_after_last_number(float_list, number):
6       """
7       Sum numbers after last number.
8
9       Parameters:
10      float_list(list): list of float
11      number(float): number after which the numbers need to be summed
12
13      Return:
14      sum(float): Sum of numbers after last number
15      """
16      sum = 0
17      if(number in float_list):
18          index = len(float_list) - float_list[-1::-1].index(number) - 1
19          for item in float_list[index:]:
20              sum += item
21      return sum
22
23  def count_pos_even_elements(float_list):
24      """
25      Return count of positive even elements.
26
27      Parameters:
28      float_list(list): list of float
29
30      Return:
31      count(int): count of positive even elements
32      """
33      count = 0
34      for item in float_list[::2]:
35          if(item > 0):
36              count+=1
37      return count
38
39
40  def task5():
41      """
42      Perform fifth task.
43      """
44      float_list = []
45      print("This program processes list of float.")
```

```python
        size = 0
        print("Input size of list:", end = " ")
        size = input_int()
        print("Generate list - 1\nInput list - 2")
        while(True):
            print("Choose number to fill list:", end = " ")
            choose = input_int()
            match choose:
                case 1:
                    float_list = generate_float_list(size)
                    break
                case 2:
                    float_list = input_float_list(size)
                    break
                case _:
                    print("Incorrect input. Try again.")
        print(float_list)
        print(f"Sum of numbers after last 0: {sum_after_last_number(float_list, 0)}")
        print(f"Count of positive even elements: {count_pos_even_elements(float_list)}")
        print()
```