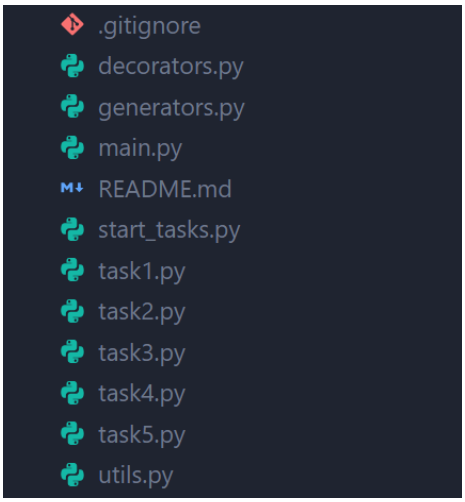
Лабораторная работа №3

Тема: стандартные типы данных, коллекции, функции, модули.
Цель: освоить базовый синтаксис языка Python, приобрести навыки работы со стандартными типами данных, коллекциями, функциями, модулями и закрепить их на примере разработки интерактивных приложений.
Выполнил: студент группы 253502 Красёв П.А.
Вариант 11.

Структура лабораторной работы:



Файл main.py

```python
from start_tasks import start_task1, start_task2, start_task3, start_task4, start_task5

# Lab3. Standard data types, collections, functions, modules.
# Performed by a student of group 253502, Krasyov Pavel.
# Fulfillment date: 21.03.2024


def main():
    task = -1
    while (True):
        print("Enter the number of task from 1 to 5, or 0 to exit:")
        try:
            task = int(input())
            if task < 0 or task > 5:
                raise Exception
        except:
            print(
                'Incorrect input, try again! Input only one number between 1 and 5, or 0 to exit\n')
            continue
        if task == 1:
            print('Task 1')
            start_task1()
            continue
        elif task == 2:
            print('Task 2')
            start_task2()
            continue
        elif task == 3:
            print('Task 3')
            start_task3()
            continue
        elif task == 4:
            print('Task 4')
            start_task4()
            continue
        elif task == 5:
            print('Task 5')
            start_task5()
            continue
        elif task == 0:
            break
    print('Program finished!')


if __name__ == '__main__':
    main()
```

Функции генераторы для создания последовательностей:

```python
import random
import string


def int_sequence(size: int) -> list[int]:
    ''' Function for generating int sequence of arbitrary size\
Numbers are generated in the interval [-150, 150]. '''
    return [random.randint(-150, 150) for i in range(size)]


def literal_sequence(size: int) -> str:
    '''  Function for generating string of arbitrary size\
Using all printable ASCII characters. '''
    return ''.join(random.choices(string.printable, k=size))


def float_sequence(size: int) -> list[float]:
    ''' Function for generating float sequence of arbitrary size\
Numbers are generated in the interval [-10, 10] with 2 decimal places. '''
    return [float(format(random.uniform(-10, 10), '.2f')) for i in range(size)]

```

Декораторы, используемые в лабораторной:

```python
from utils import stopFunction


def task1Decorator(func) -> None:
    ''' Decorator for displaying task1 info. '''
    def printInfo():
        help(func)
        print("Functiuon: ln((x+1)/(x-1))")
        print("Tailor's row: 2*sum(1/(2*n+1)*x^(2n+1), n=0..inf)\n")
        func()
    return printInfo


def chooseInputType(func) -> None:
    ''' Decorator for checking the choice between 1 and 2 (Input by keyboard or random generator) and running task. '''
    def inputType():
        help(func)
        while (True):
            while True:
                print("Enter your input type:\n1 - Keyboard\n2 - Random generator\n")
                try:
                    choice = int(input())
                    if choice != 1 and choice != 2:
                        raise Exception
                except:
                    print(
                        'Incorrect input, try again! Input only one number\n1 - Keyboard\n2 - Random generator\n')
                    continue
                break
            res = func(choice)
            if res is None and stopFunction() != "0":
                print()
                continue
            break
        print('Task exit\n')
    return inputType
```

Функции проверки ввода:

```python
def stopFunction() -> str:
    ''' Function stops running task. '''
    return input('Do you want to continue? (0 -> No | anything else -> Yes): ')


def validateNumberInput(inputMessage: str, errorMessage: str, numberType, condition=lambda x: type(x) == str) -> int:
    ''' Function validates input of a number with a given type and condition. '''
    while True:
        try:
            N = numberType(
                input(f'Enter {inputMessage}: '))
            if condition(N):
                print(f'Wrong input -> ({errorMessage})! Try again.\n')
                continue
        except ValueError:
            print('Wrong input! Try again.\n')
            continue
        break
    return N


def validateSequenceSize(name: str) -> int:
    ''' Function validates input of size of sequence. '''
    return validateNumberInput(
        f'size of {name}', 'size > 0', int, condition=lambda x: x <= 0)
```

Задание 1. В соответствии с заданием своего варианта составить программу для вычисления значения функции с помощью разложения функции в степенной ряд. Задать точность вычислений eps. Предусмотреть максимальное количество итераций, равное 500.

$$ln((x+1)/(x-1)) = 2 * \sum_{n=0}^{inf} 1/((2n+1) * x^{2n+1}), \ |x|>1$$

Функция, запускающая задание:

```python
@task1Decorator
def start_task1() -> None:
    ''' A program for calculating the value of a function using a Tailor's row of the function
with a given accuracy. '''

    while True:
        eps, x = task1.validateInput()
        if eps == 0:
            break
        task1.displayResult(x, eps)
        if stopFunction() != "0":
            print()
            continue
        break
    print('Task exit\n')
```

Функции проверки ввода, подсчёта суммы ряда Тейлора и вывода границ таблицы:

```python
from math import log
from utils import validateNumberInput


def validateInput() -> tuple[float, float]:
    ''' Function for validation input of accuracy and x value. '''
    eps = validateNumberInput(
        'accuracy (0 < acc < 1) (0 -> Exit function)', '0 < acc < 1', float, condition=lambda x: x < 0 or x >= 1)
    if eps == 0:
        return eps, 0
    x = validateNumberInput(
        'x (|x| > 1)', '|x| must be bigger than 1', float, condition=lambda x: abs(x) <= 1)
    return eps, x


def countTailor(x: float, eps: float, mathResult: float) -> tuple[float, int]:
    ''' Function that counts sum of Tailor's row with the required accuracy. '''
    tailorSum = 0
    N = -1
    while N + 1 < 500:
        N += 1
        tailorSum += 1/((2*N+1)*x**(2*N+1))
        if (abs(2*tailorSum - mathResult) >= eps/10):
            continue
        break
    return 2*tailorSum, N+1


def printBorders(type: str, length: int) -> None:
    ''' Functions prints result table borders. '''
    for i in range(5):
        if i == 0:
            print("┌─" + "─" * length,
                  end="─┬─") if type == 'top' else print("\n├─" + "─" * length, end="─┼─") \
                if type == 'mid' else print("\n└─" + "─" * length, end="─┴─")
        elif i != 4:
            print("─" * (length + 3), end="─┬─") if type == 'top' else print("─" * (length + 3), end="─┼─") \
                if type == 'mid' else print("─" * (length + 3), end="─┴─")
        else:
            print("─" * (length + 3), end="─┐\n") if type == 'top' else print("─" * (length + 3), end="─┤\n") \
                if type == 'mid' else print("─" * (length + 3), end="─┘\n")

```

Основная функция задания:

```python
def displayResult(x: float, eps: float) -> None:
    ''' Function displays th result. '''
    mathResult = log((x+1)/(x-1))
    tailorSum, iterations = countTailor(x, eps, mathResult)
    length = max(len(str(tailorSum)), len(str(iterations)),
                 len((str(eps))), len(str(x)), 9)
    data = [["x", "n", "F(x)", "Math F(x)", "eps"], [
        x, iterations, round(tailorSum, len(str(eps))), round(mathResult, len(str(eps))), eps]]
    printBorders('top', length)
    for i in range(2):
        for j in range(5):
            if j == 0:
                print("|  " + str(data[i][j]) + " " *
                      (length - len(str(data[i][j]))), end="  |  ",)
            else:
                print(str(data[i][j]).format() + " " * (length + 3 -
                      len(str(data[i][j]))), end="  |  ")
        printBorders('mid', length) if i != 1 else printBorders('bot', length)
```

Пример использования:

```
PS D:\IGI\253502_KRASYOV_11\IGI\LR3> python main.py
Enter the number of task from 1 to 5, or 0 to exit:
1
Task 1
Help on function start_task1 in module start_tasks:

start_task1() -> None
    A program for calculating the value of a function using a Tailor's row of the function
    with a given accuracy.

Functiuon: ln((x+1)/(x-1))
Tailor's row: 2*sum(1/(2*n+1)*x^(2n+1), n=0..inf)

Enter accuracy (0 < acc < 1) (0 -> Exit function): 0.0001
Enter x (|x| > 1): 2
```

x	n	F(x)	Math F(x)	eps
2.0	7	1.098607	1.098612	0.0001

```
Do you want to continue? (0 -> No | anything else -> Yes): yes

Enter accuracy (0 < acc < 1) (0 -> Exit function): 0.001
Enter x (|x| > 1): 1.125
```

x	n	F(x)	Math F(x)	eps
1.125	31	2.83313	2.83321	0.001

```
Do you want to continue? (0 -> No | anything else -> Yes): 0
Task exit

Enter the number of task from 1 to 5, or 0 to exit:
0
Program finished!
PS D:\IGI\253502_KRASYOV_11\IGI\LR3>
```

Задание 2. В соответствии с заданием своего варианта составить программу для нахождения суммы последовательности чисел. Организовать цикл, который принимает целые числа и вычисляет наименьшее из них

Функция, запускающая задание:

```python
@chooseInputType
def start_task2(inputType: int) -> None:
    ''' Program for finding the sum of a sequence of integer numbers and minimal number. '''

    if (inputType == 1):
        print('Filling an array (0 -> stop enter):')
        inputArray = task2.validateInput()
    else:
        inputArray = int_sequence(validateSequenceSize('sequence'))
    task2.displayResults(inputArray)
```

Функция проверки ввода и основная функция задания:

```python
from utils import validateNumberInput


def validateInput() -> list[int]:
    ''' Function for validation input of sequence. '''
    collection: list[int] = list()
    while True:
        n = validateNumberInput(
            'the integer number', '', int)
        if n == 0:
            break
        collection.append(n)
        continue
    return collection


def displayResults(inputArray: list[int]) -> None:
    ''' Function analising sequence and displays results. '''
    print('Input array:')
    for i in range(len(inputArray)):
        print(f'Element #{i + 1} -> {inputArray[i]}')
    if (len(inputArray) > 0):
        print(f"\nSum of elements: {sum(inputArray)}")
        print(f"Minimal value: {min(inputArray)}")
    else:
        print('Array is empty!')
```

Пример использования:

```
PS D:\IGI\253502_KRASYOV_11\IGI\LR3> python main.py
Enter the number of task from 1 to 5, or 0 to exit:
2
Task 2
Help on function start_task2 in module start_tasks:

start_task2(inputType: int) -> None
    Program for finding the sum of a sequence of integer numbers and minimal number.

Enter your input type:
1 - Keyboard
2 - Random generator

1
Filling an array (0 -> stop enter):
Enter the integer number: 1
Enter the integer number: 2
Enter the integer number: 3
Enter the integer number: -4
Enter the integer number: 5
Enter the integer number: -6
Enter the integer number: 0
Input array:
Element #1 -> 1
Element #2 -> 2
Element #3 -> 3
Element #4 -> -4
Element #5 -> 5
Element #6 -> -6

Sum of elements: 1
Minimal value: -6
Do you want to continue? (0 -> No | anything else -> Yes): 1

Enter your input type:
1 - Keyboard
2 - Random generator

2
Enter size of sequence: 6
Input array:
Element #1 -> -110
Element #2 -> -4
Element #3 -> 146
Element #4 -> -121
Element #5 -> 77
Element #6 -> 70

Sum of elements: 58
Minimal value: -121
Do you want to continue? (0 -> No | anything else -> Yes): 0
Task exit

Enter the number of task from 1 to 5, or 0 to exit:
0
Program finished!
PS D:\IGI\253502_KRASYOV_11\IGI\LR3>
```

Задание 3. В соответствии с задание своего варианта составить программу для анализа текста, вводимого с клавиатуры. В строке, введенной с клавиатуры, подсчитать количество заглавных гласных английских букв.

Функция, запускающая задание:

```python
@chooseInputType
def start_task3(inputType: int) -> None:
    ''' Program for analyzing text entered from the keyboard: counts the number of capital vowels. '''

    if (inputType == 1):
        inputText = input('Input your string. (Type `stop` to exit): ')
    else:
        inputText = literal_sequence(validateSequenceSize('string'))
    if inputText.lower() == 'stop':
        return 'stop'
    task3.displayResult(inputText)
```

Функции форматирования введенной строки, подсчёта букв и вывода:

```python
def reformString(inputText: str) -> str:
    terminators = ['\n', '\r', '\t', '\0', '\x0a',
                   '\x0b', '\x0c', '\x0d', '\x08', '\x09', '\x1b']
    for el in terminators:
        inputText = inputText.replace(el, ' ')
    return inputText


def countInputVowels(inputText: str) -> tuple[dict[str, int], dict[str, list[int]]]:
    ''' Function counts ammout of each vowel and their positions in the input string. '''
    vowels = ['A', 'E', 'I', 'O', 'U']
    countVowels: dict[str, int] = dict.fromkeys(vowels, 0)
    positions: dict[str, list[int]] = {v: list() for v in vowels}
    for i in range(len(inputText)):
        if inputText[i] in vowels:
            countVowels[inputText[i]] += 1
            positions[inputText[i]].append(i)
    return countVowels, positions


def displayPositions(inputText: str, positions: list[int]) -> None:
    ''' Function displays the positions of vowel in the string. '''
    if len(positions) == 0:
        print('No references\n')
        return
    print(inputText)
    for i in range(len(inputText)):
        if i not in positions:
            print(' ', end='')
            continue
        print('*', end='')
        positions.remove(i)
        if len(positions) == 0:
            print()
            break
```

Основная функция задания:

```python
def displayResult(inputText: str) -> str | None:
    ''' Fucntion displays the result. '''
    print('Input string:')
    print(repr(inputText))
    countVowels, positions = countInputVowels(inputText)
    inputText = reformString(inputText)
    for vowel in countVowels.keys():
        print(f'Vowel: `{vowel}` -> Ammout: {countVowels[vowel]}')
        displayPositions(inputText, positions[vowel])
```

Пример использования:

```
PS D:\IGI\253502_KRASYOV_11\IGI\LR3> python main.py
Enter the number of task from 1 to 5, or 0 to exit:
3
Task 3
Help on function start_task3 in module start_tasks:

start_task3(inputType: int) -> None
    Program for analyzing text entered from the keyboard: counts the number of capital vowels.

Enter your input type:
1 - Keyboard
2 - Random generator

1
Input your string. (Type `stop` to exit): A\nA\nA\tUII\0E\rE\nFFF
Input string:
'A\\nA\\nA\\tUII\\0E\\rE\\nFFF'
Vowel: `A` -> Ammout: 3
A\nA\nA\tUII\0E\rE\nFFF
*  *  *
Vowel: `E` -> Ammout: 2
A\nA\nA\tUII\0E\rE\nFFF
              *  *
Vowel: `I` -> Ammout: 2
A\nA\nA\tUII\0E\rE\nFFF
          **
Vowel: `O` -> Ammout: 0
No references

Vowel: `U` -> Ammout: 1
A\nA\nA\tUII\0E\rE\nFFF
         *
Do you want to continue? (0 -> No | anything else -> Yes): 1

Enter your input type:
1 - Keyboard
2 - Random generator

2
Enter size of string: 20
Input string:
"\nZ44'Ir<_B>G=i1rLBeR"
Vowel: `A` -> Ammout: 0
No references

Vowel: `E` -> Ammout: 0
No references

Vowel: `I` -> Ammout: 1
 Z44'Ir<_B>G=i1rLBeR
     *
Vowel: `O` -> Ammout: 0
No references

Vowel: `U` -> Ammout: 0
No references

Do you want to continue? (0 -> No | anything else -> Yes): 0
```

Задание 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 количество слов в строке;
б) найти самое длинное слово и его порядковый номер;
в) вывести каждое нечетное слово

Функция, запускающая задание:

```python
def start_task4() -> None:
    ''' Program for parsing a string initialized in program code:\na) determine the number of words in a line;\n\
b) find the longest word and its serial number;\nc) print every odd word '''
    task4.analiseString()
    print('Task exit\n')
```

Функции очистки строки от знаков препинания, поиска самого длинного слова, вывода нечётных слов и основная функция задания:

```python
def clearString() -> list[str]:
    ''' Function converts a string to replace all non-letter characters with spaces. '''
    clearString = "".join(
        el for el in initialString if el.isalpha() or el == ' ' or el == '-')
    return clearString.split(' ')


def findMaxLenWord(words: list[str]) -> None:
    ''' Function finds max len word and its position in the string. '''
    maxLenWords: list[tuple[int, str]] = [(0, '')]
    for id, word in enumerate(words):
        if len(word) > len(maxLenWords[0][1]):
            maxLenWords.clear()
            maxLenWords.append((id, word))
        elif len(word) == len(maxLenWords[0][1]):
            maxLenWords.append((id, word))
    print(
        f"Max length word{'s' if len(maxLenWords) > 1 else '' } => Max length: {len(maxLenWords[0][1])}:")
    for id, word in maxLenWords:
        print(f'{word} -> Position: {id}')


def displayOddWords(words: list[str]) -> None:
    ''' Function prints odd words of the string. '''
    print('Odd words:')
    for i in range(0, len(words), 2):
        print(words[i], end=" ")


def analiseString() -> None:
    ''' Function analises string and displays results. '''
    print('Initial string:')
    print(initialString)
    print("\nAnalised string:")
    words = clearString()
    print(f'Ammount of words: {len(words)}')
    findMaxLenWord(words)
    displayOddWords(words)
    print('Task exit\n')
```

Пример использования:

```
PS D:\IGI\253502_KRASYOV_11\IGI\LR3> python main.py
Enter the number of task from 1 to 5, or 0 to exit:
4
Task 4
Help on function start_task4 in module start_tasks:

start_task4() -> None
    Program for parsing a string initialized in program code:
    a) determine the number of words in a line;
    b) find the longest word and its serial number;
    c) print every odd word

Initial string:
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.

Analised string:
Ammount of words: 55

Max length words => Max length: 11:
considering -> Position: 3
daisy-chain -> Position: 30

Odd words:
So was in own as as could the day her very and whether pleasure making daisy-chain be the of up picking daisies suddenly White with eyes close her Task exit

Task exit

Enter the number of task from 1 to 5, or 0 to exit:
0
Program finished!
PS D:\IGI\253502_KRASYOV_11\IGI\LR3>
```

Задание 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.

Найти количество элементов списка, лежащих в диапазоне от A до B (параметры A и B вводятся с клавиатуры пользователем) и сумму элементов списка, расположенных после максимального элемента

Функция запуска задания:

```python
@chooseInputType
def start_task5(inputType: int) -> None:
    ''' Program for processing real lists:\nthe number of list elements lying in the range from A to B (parameters A\
and B are entered from the keyboard by the user)\nand the sum of the list elements located after the maximum element '''

    if (inputType == 1):
        collection = task5.validateInputArray()
    else:
        collection = float_sequence(validateSequenceSize('sequence'))
    if collection is None:
        return 'stop'
    task5.analiseCollection(collection)
```

Функции проверки ввода последовательности и границ диапазона:

```python
def validateInputArray() -> list[float]:
    ''' Function validates input of sequence and its lenght. '''
    N = validateNumberInput(
        'length of array (len > 0) (0 -> Exit function)', 'len > 0', int, condition=lambda x: x < 0)
    if N == 0:
        return None
    k = 0
    collection = list[float]()
    while k < N:
        el = validateNumberInput(
            f'#{k + 1} element (float number)', '', float)
        collection.append(el)
        k += 1
    return collection


def validateInputBorders() -> tuple[int, int]:
    ''' Function validates input of interval borders. '''
    while True:
        A = validateNumberInput('left border: ', '', int)
        B = validateNumberInput(
            'rigth border: ', '', int)
        if B < A:
            print('Wrong input -> (left border <= right border)! Try again.')
            continue
        break
    return A, B
```

Функция вывода последовательности, поиска максимального элемента и количества элементов в диапазоне

```python
def displayCollection(collection: list[float]) -> None:
    ''' Function displays collection. '''
    print('Entered collection:')
    for el in collection:
        print(el, end=" ")
    print()


def findAmmountInInterval(collection: list[float], A: int, B: int) -> int:
    ''' Function counts an ammount of collection elements in the interval. '''
    return sum(1 if el >= A and el <= B else 0 for el in collection)


def findMaxElement(collection: list[float]) -> tuple[int, float]:
    ''' Function finds collection max element and its position. '''
    return max(enumerate(collection), key=lambda el: el[1])


def findSumInterval(collection: list[float], start: int) -> float:
    ''' Function finds sum of elements from the `start` position to the end of collection. '''
    return sum(collection[i] for i in range(start, len(collection)))
```

Основная функция задания

```python
def analiseCollection(collection: list[float]) -> str | None:
    ''' Function analises function and displays results. '''
    displayCollection(collection)
    A, B = validateInputBorders()
    print('Analised collection:')
    print(
        f'Ammount of elements in [{A}, {B}]: {findAmmountInInterval(collection, A, B)}')
    maxElement = findMaxElement(collection)
    print(f"Max element: {maxElement[1]}")
    print(
        f"Sum of elements after Max: {findSumInterval(collection, maxElement[0] + 1)}")
```

Пример использования:

```
PS D:\IGI\253502_KRASYOV_11\IGI\LR3> python main.py
Enter the number of task from 1 to 5, or 0 to exit:
5
Task 5
Help on function start_task5 in module start_tasks:

start_task5(inputType: int) -> None
    Program for processing real lists:
    the number of list elements lying in the range from A to B (parameters Aand B are entered from the keyboard by the user)
    and the sum of the list elements located after the maximum element

Enter your input type:
1 - Keyboard
2 - Random generator

1
Enter length of array (len > 0) (0 -> Exit function): 10
Enter #1 element (float number): 1.25
Enter #2 element (float number): -34.08
Enter #3 element (float number): 5.3
Enter #4 element (float number): 3.33
Enter #5 element (float number): -0.3333
Enter #6 element (float number): 20.0
Enter #7 element (float number): 9.4
Enter #8 element (float number): 2
Enter #9 element (float number): 1.6
Enter #10 element (float number): 0
Entered collection:
1.25 -34.08 5.3 3.33 -0.3333 20.0 9.4 2.0 1.6 0.0
Enter left border: : -5
Enter rigth border: : 5
Analised collection:
Ammount of elements in [-5, 5]: 6
Max element: 20.0
Sum of elements after Max: 13.0
Do you want to continue? (0 -> No | anything else -> Yes): 1
```

```
Enter your input type:
1 - Keyboard
2 - Random generator

2
Enter size of sequence: 10
Entered collection:
1.44 -6.09 1.68 -6.94 7.52 6.88 4.4 6.2 -9.33 6.37
Enter left border: : -1 1
Wrong input! Try again.

Enter left border: : -1
Enter rigth border: : 1
Analised collection:
Ammount of elements in [-1, 1]: 0
Max element: 7.52
Sum of elements after Max: 14.52
Do you want to continue? (0 -> No | anything else -> Yes): 0
Task exit

Enter the number of task from 1 to 5, or 0 to exit:
0
Program finished!
PS D:\IGI\253502_KRASYOV_11\IGI\LR3>
```