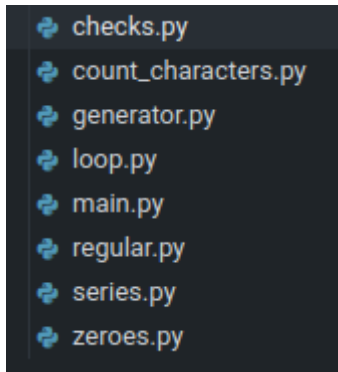


Отчёт по лабораторной работе №3. Избранные главы информатики.

Сергеев Р.В., гр. 253502.

Структура лабораторной работы:



Задание 1.

вычисление суммы ряда:

```
def calc_acos_series(x: float, eps: float):  
    """  
    calculates acos series function with eps accuracy  
    return calculated value and iterations count  
    """  
    sum = math.pi / 2;  
    part = eps * 1337;  
    n = 0;  
    while (True):  
        temp = math.factorial(2 * n) * math.pow(x, 2 * n + 1) / (math.pow(4, n) * math.pow(math.factorial(n), 2) * (2 * n + 1))  
        if (abs(part - temp) < eps):  
            break  
        part = temp  
        sum -= part  
        n = n + 1  
    return sum, n
```

основная функция задания:

```

def calc_acos(n:int, eps: float):
    '''
    calculates acos from x=-1 to 1 and prints table
    number of elements = n
    accuracy = eps
    '''
    if (n < 2):
        print("iteration count must be greater than 1")
        return
    if (eps < 0):
        print("accuracy must be greater than 0")
        return

    step = 2 / (n - 1)
    if (step > 500):
        raise Exception("iteration count more than 500")
    data = [[],[],[],[],[]]
    x = -1
    data[0].append("x")
    data[1].append("n")
    data[2].append("F(x)")
    data[3].append("Math F(x)")
    data[4].append("eps")
    while (x <= 1):
        try:
            series, n = calc_acos_series(x, eps)
        except:
            series = "calculation overflow"
            n = "inf"
        fn = math.acos(x)
        data[0].append(x.__str__())
        data[1].append(n.__str__())
        data[2].append(series.__str__())
        data[3].append(fn.__str__())
        data[4].append(eps.__str__())
        x += step
    return data

```

функция вывода таблицы:

```

def print_table(data: list[list]):
    '''
    prints table for calc_acos
    '''
    lengths = []
    for i in range(len(data)):
        lengths.append(max(len(s) for s in data[i]))
        if i == 0:
            print("┌" + "-" * lengths[i], end="┐")
        elif i != len(data) - 1:
            print("-" * (lengths[i] + 3), end="┐")
        else:
            print("-" * (lengths[i] + 3), end="└")
    print("")
    for i in range(len(data[0])):
        for j in range(len(data)):
            if j == 0:
                print("| " + data[j][i] + " " * (lengths[j] - len(data[j][i])), end=" | ",)
            else:
                print(data[j][i] + " " * (lengths[j] + 3 - len(data[j][i])), end=" | ",)
        print("")
    for j in range(len(data)):
        if i != len(data[0]) - 1:
            if j == 0:
                print("┌" + "-" * lengths[j], end="┐")
            elif j != len(data) - 1:
                print("-" * (lengths[j] + 3), end="┐")
            else:
                print("-" * (lengths[j] + 3), end="└")
        else:
            if j == 0:
                print("┌" + "-" * lengths[j], end="┐")
            elif j != len(data) - 1:
                print("-" * (lengths[j] + 3), end="┐")
            else:
                print("-" * (lengths[j] + 3), end="└")
    print("")

```

Пример выполнения:

TASK: calculate acos(x) and acos series
 Input number of elements and accuracy (n,e):
 Input integer (q - exit):
 10
 Input float number (q - exit):
 0.001

x	n	F(x)	Math F(x)	eps
-1	12	2.9781938344214005	3.141592653589793	0.001
-0.7777777777777778	6	2.460600092520032	2.4619188346815495	0.001
-0.5555555555555556	4	2.159628200514815	2.1598272970111707	0.001
-0.3333333333333337	3	1.9106111416097116	1.9106332362490186	0.001
-0.11111111111111116	2	1.6821360615914216	1.6821373411358607	0.001
0.11111111111111105	2	1.4594565919983715	1.459455312453933	0.001
0.3333333333333326	3	1.2309815119800818	1.2309594173407747	0.001
0.5555555555555555	4	0.981964453074978	0.9817653565786228	0.001
0.7777777777777777	6	0.6809925610697607	0.6796738189082441	0.001
0.9999999999999999	12	0.16339881916839327	1.4901161193847656e-08	0.001

Задание 2.

```
def numloop() -> float:
    """
    get numbers from input and calculate multiplication of last digits
    """
    sum = 1
    while(True):
        num = inputInt()
        if (num == 0):
            break
        sum *= num % 10
    return sum
```

Пример выполнения:

```
TASK: calculate multiplication of last digits
Input numbers, 0 - end input
Input integer (q - exit):
12
Input integer (q - exit):
13
Input integer (q - exit):
0
результат = 6
```

Задание 3.

```
def count_characters(str: str):  
    '''  
    count spaces, digits and punctuation characters in string  
    '''  
    data = {"spaces": 0, "digits": 0, "punctuation": 0}  
    punctuation = [",", ".", ";", ":", "-"]  
    for c in str:  
        if c == " "  
            data["spaces"] += 1  
        elif ord(c) > 47 and ord(c) < 58:  
            data["digits"] += 1  
        elif c in punctuation:  
            data["punctuation"] += 1  
    return data.items()
```

Пример выполнения:

```
TASK: calculate count of spaces, digits and punctuation characters  
Input string:  
aboba 123 ;;; bebra, bob  
spaces: 4  
digits: 3  
punctuation: 4
```

Задание 4.

```
def words_with_vowel(str: str) -> list[str]:  
    '''  
    return array of strings where all elements starts or ends with vowel  
    '''  
  
    word_list = []  
    vowels = ["e", "y", "u", "i", "o", "a"]  
    word = ""  
    for c in str:  
        if (ord(c) > 64 and ord(c) < 91) or (ord(c) > 96 and ord(c) < 132):  
            word += c  
        else:  
            if len(word) > 0 and (word[0] in vowels or word[-1] in vowels):  
                word_list.append(word)  
            word = ""  
    return ", ".join(word_list)
```

```
def all_characters(str: str):  
    '''  
    count all characters in string  
    '''  
  
    symbols = {}  
    for c in str:  
        if c not in symbols:  
            symbols[c] = 0  
        symbols[c] += 1  
    for k, v in symbols.items():  
        print(f"{k}: {v}")
```

```
def alphabet_after_comma(str: str) -> list[str]:  
    '''  
    return array of strings in alphabet order with elements, with stand after comma  
    '''  
  
    word_list = []  
    word = ""  
    comma_found = False  
    word_starts = False  
    for c in str:  
        if c == ",":  
            comma_found = True  
        if comma_found:  
            if (ord(c) > 64 and ord(c) < 91) or (ord(c) > 96 and ord(c) < 132):  
                word += c  
                word_starts = True  
            elif word_starts:  
                comma_found = False  
                word_starts = False  
                word_list.append(word)  
                word = ""  
    word_list.sort()  
    return ", ".join(word_list)
```

Пример выполнения:

```
Origin 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  
up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.  
  
All words, with vowel at start or end of it:  
So, she, in, own, as, as, she, the, day, made, very, sleepy, and, the, pleasure, of, a, daisy, be, the, trouble, of, up, and, the, suddenly, a, White, eyes, close, by  
  
How many times each character repeats:  
S: 1  
o: 12  
.: 54  
s: 15  
h: 17  
e: 31  
w: 8  
b: 16  
c: 5  
n: 15  
i: 17  
d: 13  
r: 11  
g: 5  
m: 3  
,: 4  
l: 10  
u: 7  
f: 4  
t: 14  
y: 7  
v: 1  
p: 6  
k: 3  
-: 1  
b: 5  
W: 1  
R: 1  
,: 1  
None  
  
All words after comma:  
as, for, when, whether
```

Задание 5.

```

def getSublist(arr: list[float]):
    '''
    return array with elements between first two zeroes
    '''
    beg = 0
    end = 0
    for i in range(len(arr)):
        if arr[i] == 0:
            if beg != 0:
                end = i
                break
            beg = i
    return arr[beg + 1:end]

def max_elem(arr: list[float]):
    '''
    return max element of array
    '''
    return max(arr)

def multiply(arr: list[float]):
    '''
    return multiplication of array elements
    '''
    res = 1
    for elem in arr:
        res *= elem
    return res

```

Пример выполнения:

```

TASK: find max element and multiplication of elements between first two zero elements
Generate elements? (Y) or input (n)? (Y / n):
Y
440, -607, 0, 675, -440, -209, 0, 773, 0, 0,
Max element: 675
Multiplication of elements = 62073000

```


Дополнительные функции:

```
def inputInt() -> int:
    '''
    get integer from stdin, checks if it integer
    '''
    val: str
    try:
        print("Input integer (q - exit):")
        val = input()
        return int(val)
    except:
        if val == "q":
            os._exit()
        print("Incorrect input. Try again:")
        return inputInt()
```

```
def inputFloat() -> float:
    '''
    get float from stdin, checks if it float
    '''
    val: str
    try:
        print("Input float number (q - exit):")
        val = input()
        return float(val)
    except:
        if val == "q":
            os._exit()
        print("incorrect input. :")
        return inputFloat()
```

```
def inputZeroes() -> list[float]:
    '''
    get array of floats from stdin, checks if elements are float and if there are 2 zeroes
    '''
    print("input elements, e - break:")
    arr = []
    while (True):
        val = input()
        if val == "e":
            break
        try:
            arr.append(float(val))
        except:
            print("icorrect input")
    if arr.count(0) < 2:
        print("must be at least 2 zero values")
        return inputZeroes()
    return arr
```

