

Отчет о проделанной работе

Задание 1.

Осуществить набор, анализ следующих программ:

- Нахождение максимума двух чисел
- Нахождение максимума из 3 чисел
- Определение количества цифр в целом числе
- Нахождение всех двузначных чисел, в которых есть заданная цифра

Результат: Набор осуществлен -> task_1.py

Задание 2.

Задача 1.

Даны три целых числа. Определить у какого числа больше сумма цифр. Вывод результата предусмотреть в основной программе. Расчет суммы цифр оформить в функции.

```
def count_digits(x):
    if type(x) != int:
        raise ValueError(f'Число {x} не целое или не является числом.
Проверьте вводимые типы данных')
    sum_digits = 0
    for digit in str(x):
        sum_digits += int(digit)
    return sum_digits

def max_sum_digits(a, b, c):
    max_result = 0
    max_elem = a
    for index, elem in enumerate([a,b,c]):
        if count_digits(elem) > max_result:
            max_result = count_digits(elem)
            max_elem = elem
    return max_elem

#print(max_sum_digits(1,2,'koshka')) #Число koshka не целое или не является
числом. Проверьте вводимые типы данных

print(max_sum_digits(123,245,999)) #999

print(max_sum_digits(000000,1000,2)) #2
```

Задача 2.

Рассчитать и вывести периметр и площадь прямоугольника. Расчеты оформить в функции.

```
def rectangle_parameters(a,b):
    if type(a) not in (int, float):
        raise ValueError(f'{a} не является вещественным числом')
    if type(b) not in (int, float):
        raise ValueError(f'{b} не является вещественным числом')

    print(f'Периметр прямоугольника: {2*(a+b)}\n\
Его площадь: {a*b}')

#rectangle_parameters('banan', 4) #banan не является вещественным числом
#rectangle_parameters(5, 'koshka') # koshka не является вещественным числом

rectangle_parameters(1,2)
rectangle_parameters(14.3, 5)
```

Задача 3.

Написать программу, подсчитывающую количество цифр числа, используя для этого функцию.

```
def count_digits(x):
    if type(x) != int:
        raise ValueError(f'Число {x} не целое или не является числом.
Проверьте вводимые типы данных')
    sum_digits = 0
    for digit in str(x):
        sum_digits += 1
    return sum_digits

#print(count_digits('koshka')) # Число koshka не целое или не является числом.
#Проверьте вводимые типы данных

print(count_digits(281234))
```

Задание 3. (вариант 4)

Задача 1.

Найти сумму чисел ряда 1,2,3,...,60 с использованием функции нахождения суммы. Использовать локальные переменные.

Создадим список 1,2,3,...,60

```
my_list = [i for i in range(1,60 + 1)]

print(f'Сумма ряда от 1 до 60: {sum(my_list)}') # Сумма ряда от 1 до 60: 1830
```

Задача 2.

Описать функцию RectPS(x1,y1,x2,y2,P,S), вычисляющую периметр P и площадь S прямоугольника со сторонами, параллельными осям координат, по координатам (x1, y1), (x2, y2) его противоположных вершин (x1, y1, x2, y2 – входные, P и S – выходные параметры вещественного типа). С помощью этой функции найти периметры и площади трех прямоугольников с данными противоположными вершинами.

```
def RectPS(x1,x2,y1,y2):
    for elem in [x1,x2,y1,y2]:
        if type(elem) not in (int, float):
            raise ValueError(f'{elem} не является вещественным числом')
    a = abs(x1-x2)
    b = abs(y1-y2)
    print(f'Периметр прямоугольника: {2*(a+b)}\n\
Его площадь: {a*b}')

#RectPS('banan',1,2,3) #ValueError: banan не является вещественным числом
# RectPS(4,'koshka',2,3) #ValueError: koshka не является вещественным числом

RectPS(1,2,3,4)
RectPS(-1,-2,3,4)
RectPS(16.6,92,-38,44)

Периметр прямоугольника: 4

Его площадь: 1

Периметр прямоугольника: 4

Его площадь: 1

Периметр прямоугольника: 314.8

Его площадь: 6182.8
```