

Практическое занятие № 5

Тема: составление программ с функциями в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с функциями в IDE PyCharm Community.

Постановка задачи.

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

Тип алгоритма:

Циклический, ветвящийся.

Текст программы:

"Найти сумму чисел ряда 1,2,3,...,60 с использованием функции нахождения суммы.

Использовать локальные переменные."

```
def generate():
```

```
    result = sum(range(1, 61))
```

```
    return result
```

```
result = generate()
```

```
print(f'Сумма чисел ряда 1,2,3,...,60 равняется - {result}')
```

Протокол работы программы:

Сумма чисел ряда

1,2,3,...,60 равняется -

1830

Process finished with exit

code 0

2. Описать функцию RectPS(x_1, y_1, x_2, y_2, P, S), вычисляющую периметр P и площадь S прямоугольника со сторонами, параллельными осям координат, по координатам (x_1, y_1) , (x_2, y_2) его противоположных вершин (x_1, y_1, x_2, y_2 — входные, P и S — выходные параметры вещественного типа). С помощью этой функции найти периметры и площади трех прямоугольников с данными противоположными вершинами.

Тип алгоритма:

Циклический, ветвящийся.

Текст программы:

"Описать функцию RectPS(x_1, y_1, x_2, y_2, P, S),
вычисляющую периметр P и площадь S
прямоугольника со сторонами, параллельными осям
координат, по координатам $(x_1,$
 $y_1)$, (x_2, y_2) его противоположных вершин ($x_1, y_1, x_2,$
 y_2 — 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):

    length = abs(x2 - x1) # Находим длину

    width = abs(y2 - y1) # Находим ширину

    P = 2 * (length + width)

    S = length * width

    return P, S

# Обработка исключений

while True:

    try:

        # Ввод значений для 1 прямоугольника

        x1, x2, y1, y2 = [int(input(f'Введите значение для
{i}: ')) for i in ['x1', 'x2', 'y1', 'y2']]

        break

    except ValueError:

        print('Ошибка: Введите корректные числа')
```

заново!')

P, S = RectPS(x1, x2, y1, y2)

print(fПериметр 1 прямоугольника равен - {P},
площадь - {S}')

#####

Обработка исключений

while True:

try:

Ввод значений для 2 прямоугольника

x1, x2, y1, y2 = [int(input(fВведите значение для
{i}: ')) for i in ['x1', 'x2', 'y1', 'y2']]

break

except ValueError:

print('Ошибка: Введите корректные числа
заново!')

```
P, S = RectPS(x1, x2, y1, y2)
```

```
print(fПериметр 2 прямоугольника равен - {P},  
площадь - {S}')
```

```
#####  
#####  
#####
```

```
# Обработка исключений
```

```
while True:
```

```
    try:
```

```
        # Ввод значений для 3 прямоугольника
```

```
        x1, x2, y1, y2 = [int(input(fВведите значение для  
{i}: ')) for i in ['x1', 'x2', 'y1', 'y2']]
```

```
        break
```

```
    except ValueError:
```

```
        print('Ошибка: Введите корректные числа  
заново!')
```

```
P, S = RectPS(x1, x2, y1, y2)
```

```
print(fПериметр 3 прямоугольника равен - {P},  
площадь - {S}')
```

```
#####  
#####  
#####
```

Протокол работы программы:

Введите значение для x1: фгв=фoыв

Ошибка: Введите корректные числа заново!

Введите значение для x1: 3

Введите значение для x2: 6

Введите значение для y1: 2

Введите значение для y2: 7

Периметр 1 прямоугольника равен - 16, площадь - 15

Введите значение для x1: 2

Введите значение для x2: 4

Введите значение для y1: 1

Введите значение для y2: 5

Периметр 2 прямоугольника равен - 12, площадь - 8

Введите значение для x1: 2

Введите значение для x2: 5

Введите значение для y1: 1

Введите значение для y2: 5

Периметр 3 прямоугольника равен - 14, площадь - 12

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ циклической, ветвящейся структуры в IDE PyCharm Community. Были использованы языковые конструкции while, if, try/except, for, range, sum(), def _ (), abs(). Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.