

## Отчет

### Практическое занятие № 4

**Тема:** составление программ циклической структуры в IDE PyCharm Community.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community.

**Постановка задачи.**

1. Дано целое число  $N (>0)$ . Найти произведение  $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$  ( $N$  сомножителей).
2. Даны положительные числа  $A, B, C$ . На прямоугольнике размера  $A \times B$  размещено максимально возможное количество квадратов со стороной  $C$  (без наложений). Найти количество квадратов, размещенных на прямоугольнике. Операции умножения и деления не использовать.

**Тип алгоритма:** циклический

**Блок-схема алгоритма:**



### Текст программы:

```
# Дано целое число N (>0). Найти произведение 1.1 * 1.2 * 1.3 *... (N сомножителей).

try:
    # Запрашиваем у пользователя ввод целого числа N
    N = int(input("Введите целое число: "))

    # Проверяем, что N больше 0
    if N <= 0:
        raise ValueError("Число должно быть больше 0.")

    # Инициализируем переменную для произведения
    product = 1.0

    # Вычисляем произведение от 1.1 до 1.0 + N
    for i in range(1, N + 1):
        product *= (1 + i / 10) # 1.1, 1.2, ..., 1.0 + N

    # Выводим результат
    print(f"Произведение первых {N} сомножителей равно: {product}")

except ValueError as e: # Обработка исключений, если ввод некорректен
    print(f"Ошибка: {e}!")
```

### Протокол работы программы:

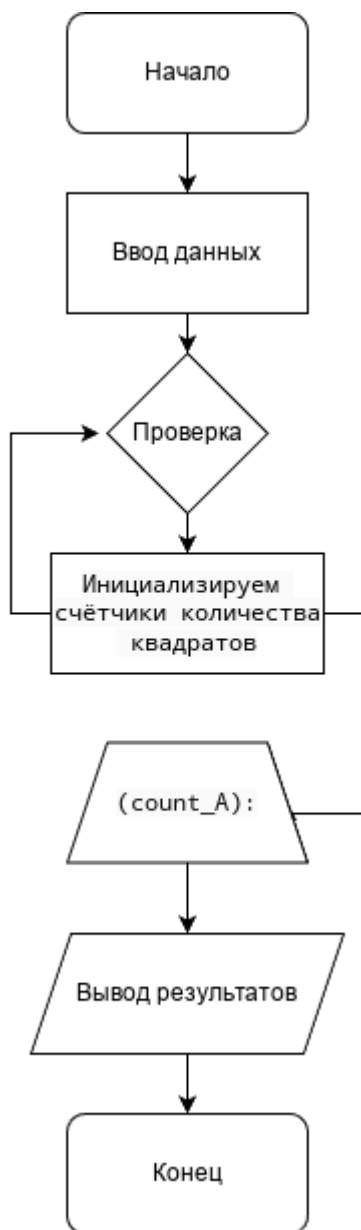
Введите целое число: 123

Произведение первых 123 сомножителей равно: 4.098519362022393e+96

Process finished with exit code 0

**Тип алгоритма:** циклический

**Блок-схема алгоритма:**



Студент группы ИС-27 Филонов.Н.А.

**Текст программы:**

```

# Даны положительные числа A, B, C. На прямоугольнике размера A x B размещено
# максимально возможное количество квадратов со стороной C (без наложений).
# Найти количество квадратов, размещенных на прямоугольнике. Операции
# умножения и деления не использовать.

# Вводим значения сторон прямоугольника и квадрата
try:

    A, B, C = float(input("Введите длину прямоугольника A: ")), float(input("Введите ширину прямоугольника B: ")), \
        float(input("Введите сторону квадрата C: "))

    # Проверяем, что все введённые числа положительные
    if A <= 0 or B <= 0 or C <= 0:
        raise ValueError("Все значения должны быть положительными числами")

    # Инициализируем счётчики количества квадратов
    count_A = 0 # Количество квадратов по стороне A
    count_B = 0 # Количество квадратов по стороне B

    # Подсчёт количества квадратов вдоль длины A
    remaining_length = A
    while remaining_length >= C:
        count_A += 1
        remaining_length -= C # Вычитаем сторону квадрата

    # Подсчёт количества квадратов вдоль ширины B
    remaining_width = B
    while remaining_width >= C:
        count_B += 1
        remaining_width -= C # Вычитаем сторону квадрата

    # Общее количество квадратов
    total_squares = 0
    for _ in range(count_A):
        total_squares += count_B # Добавляем количество квадратов вдоль ширины

    print(f"Максимально возможное количество квадратов на прямоугольнике: {total_squares}")

except ValueError as ve:
    print(f"Ошибка ввода: {ve}")
except Exception as e:
    print(f"Произошла ошибка: {e}")

```

## Протокол работы программы:

Введите длину прямоугольника A: 123

Введите ширину прямоугольника B: 123

Введите сторону квадрата C: 123

Максимально возможное количество квадратов на прямоугольнике: 1

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 in, range. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.