

Практическое занятие №4

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.

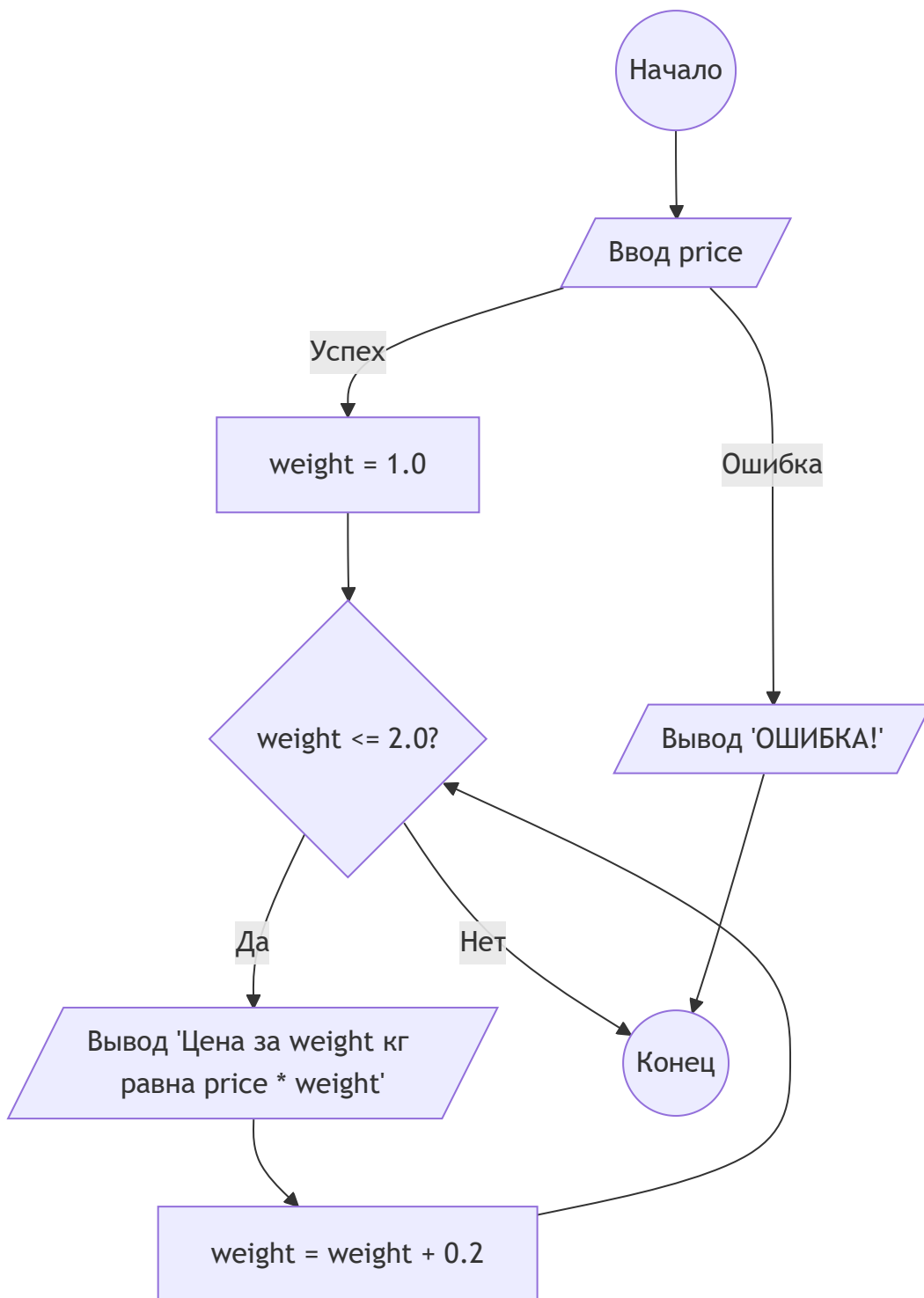
Задание №1

Постановка задачи:

Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1.2, 1.4, ..., 2 кг конфет.

Тип алгоритма: циклический.

Блок-схема алгоритма:



Текст программы:

```
try:
    price = int(input("Введите цену за 1 кг конфет: "))
    weight = 1.0
    while weight <= 2:
        print("Цена за ", weight, "равна ", price * weight)
```

```
        weight += 0.2
except ValueError:
    print("ОШИБКА!")
```

Протокол работы программы (примеры):

Введите цену за 1 кг конфет: 100

Цена за 1 кг равна 100

Цена за 1 кг равна 120

Цена за 1 кг равна 140

Цена за 1 кг равна 160

Цена за 1 кг равна 180

Цена за 2 кг равна 200

Введите цену за 1 кг конфет: сто # неправильный ввод
ОШИБКА!

Введите цену за 1 кг конфет: 150

Цена за 1 кг равна 150

Цена за 1 кг равна 180

Цена за 1 кг равна 210

Цена за 1 кг равна 240

Цена за 1 кг равна 270

Цена за 2 кг равна 301

Вывод:

В ходе выполнения практического задания были закреплены навыки использования цикла `while`, форматирования вывода с помощью f-строк и обработки исключений для корректной работы программы при неверном вводе данных. Также добавлен вывод веса с одним знаком после запятой и цены с двумя знаками для большей наглядности и соответствия денежному формату.

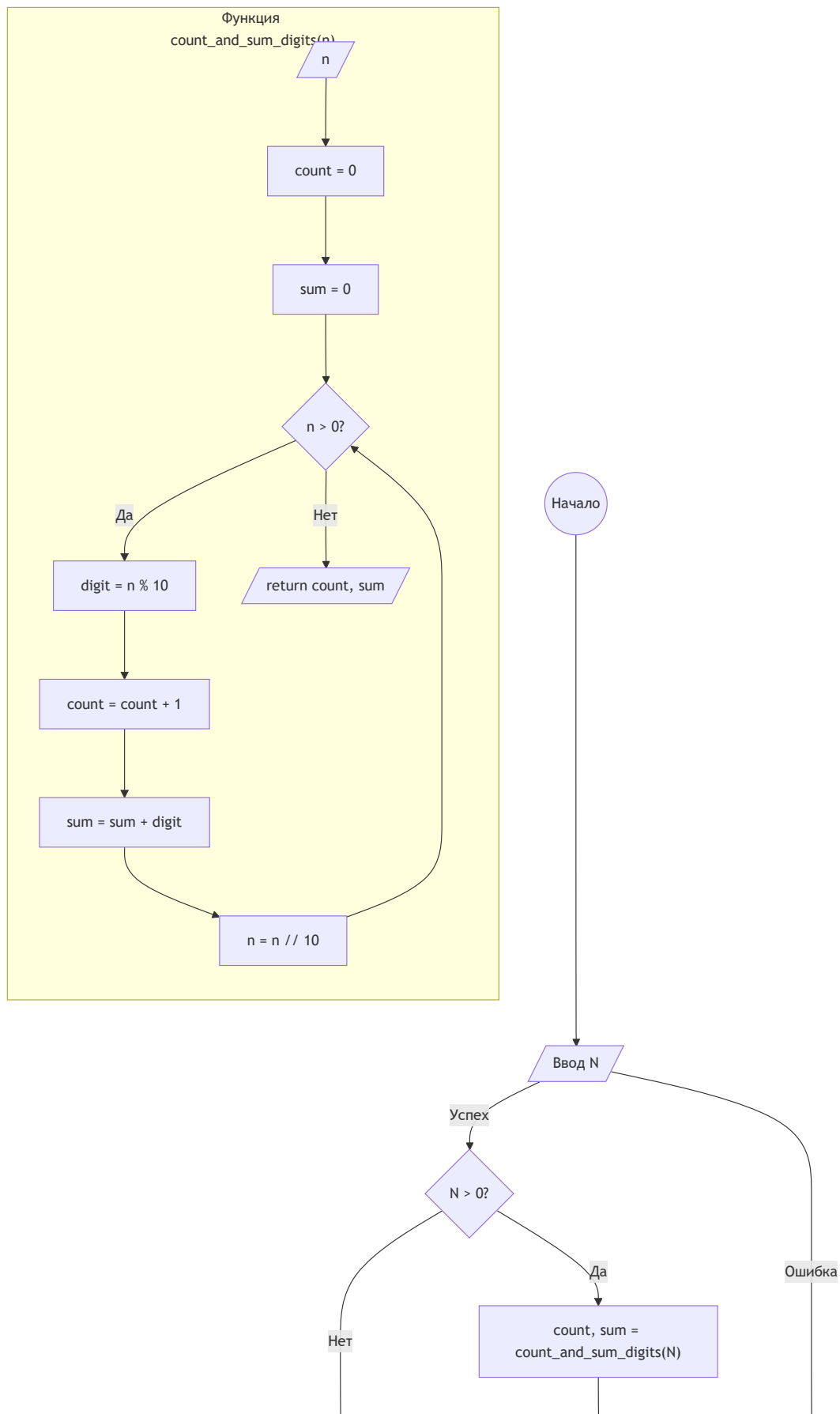
Задание №2

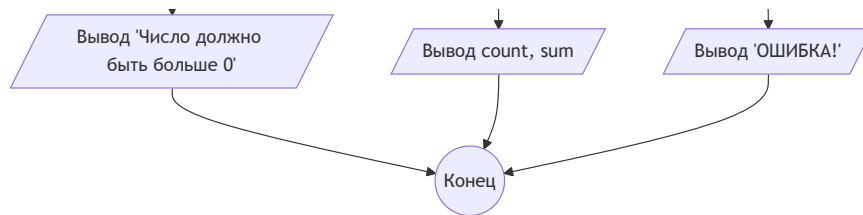
Постановка задачи:

Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.

Тип алгоритма: циклический.

Блок-схема алгоритма:





Текст программы:

```
def count_and_sum_digits(n):
    count = 0
    total_sum = 0

    while n > 0:
        last_digit = n % 10
        count += 1
        total_sum += last_digit
        n //= 10
    return count, total_sum

try:
    n = int(input("Введите целое число N (> 0): "))
    if n > 0:
        print(count_and_sum_digits(n))
        digit_count, digit_sum = count_and_sum_digits(n)
        print(f"Количество цифр: {digit_count}")
        print(f"Сумма цифр: {digit_sum}")
    else:
        print("Число должно быть больше 0.")
except ValueError:
    print("ОШИБКА!")
```

Протокол работы программы (примеры):

Введите целое число N (> 0): 12345
Количество цифр: 5
Сумма цифр: 15

Введите целое число N (> 0): 0
Число должно быть больше 0.

Введите целое число N (> 0): -123
Число должно быть больше 0.

Введите целое число N (> 0): abc
ОШИБКА!

Введите целое число N (> 0): 9999
Количество цифр: 4
Сумма цифр: 36

Вывод:

В ходе выполнения практического задания были закреплены навыки использования циклов `while`, функций, операций деления нацело (`//`) и взятия остатка от деления (`%`) для решения задачи определения количества и суммы цифр числа.