

**ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №1  
«Условные операторы и циклы в языке Python»  
по дисциплине «Введение в системы искусственного интеллекта»  
Вариант 11**

Выполнил: студент группы ИВТ-б-о-18-1  
Солдатенко Евгений Михайлович

\_\_\_\_\_ (подпись)

Проверил:  
доцент кафедры инфокоммуникаций,  
Воронкин Роман Александрович

\_\_\_\_\_ (подпись)

Ставрополь, 2022

**Цель работы:** приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

### Ход работы:

Была решена задача 1 согласно варианта, составлена UML-диаграмма деятельности и программа с использованием конструкций ветвления. Номер варианта – 11.

Компания по снабжению электроэнергией взимает плату с клиентов по тарифу:

7 р. за 1 кВт/ч за первые 250 кВт/ч;

17 р. за кВт/ч, если потребление свыше 250, но не превышает 300 кВт/ч;

20 р. за кВт/ч, если потребление свыше 300 кВт/ч.

Потребитель израсходовал  $n$  кВт/ч. Подсчитать плату.

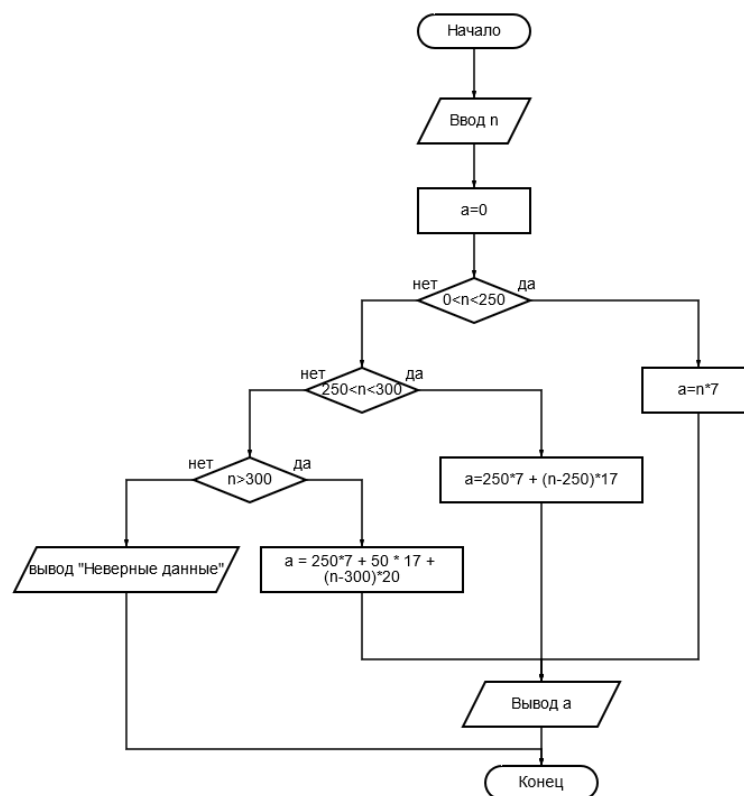


Рисунок 1 – UML-диаграмма задачи 1

```

n = int(input("Использовано кВт/ч: "))
a = 0

if n >= 0 and n <= 250:
    a = n*7

elif n > 250 and n <= 300:
    a = (250*7 + (n-250)*17)

elif n > 300:
    a = 250*7 + 50 * 17 + (n-300)*20
else:
    print("Неверно введенные данные")

print(f"Плата за использованную электроэнергию: {a}")

```

Использовано кВт/ч: 400  
3е условие  
Плата за использованную электроэнергию: 4600

Рисунок 2 – Решение задачи 1

Была решена задача 2 согласно варианта, составлена UML-диаграмма деятельности и программа с использованием конструкций ветвления. Номер варианта – 11.

Определить, есть ли среди трёх заданных чисел чётные.

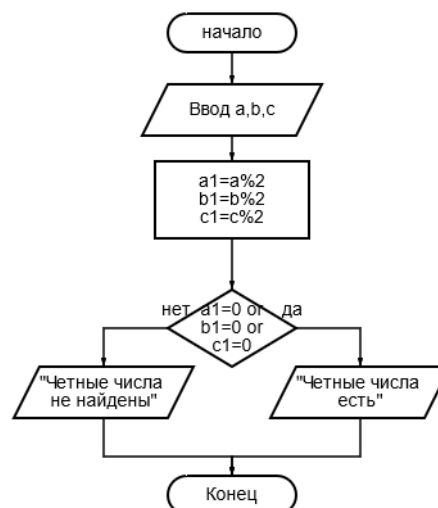


Рисунок 3 – UML-диаграмма задачи 2

```

a = int(input("Введите первое число: "))
b = int(input("Введите второе число: "))
c = int(input("Введите третье число: "))

a1 = a % 2
b1 = b % 2
c1 = c % 2

if a1 == 0 or b1 == 0 or c1 == 0:
    print("Четные числа есть")
else: print("Четные числа не найдены")

```

Введите первое число: 1  
Введите второе число: 2  
Введите третье число: 9  
Четные числа есть

Рисунок 4 – Решение задачи 2

Была составлена UML-диаграмма деятельности и программа с использованием конструкций цикла для решения задачи 3. Номер варианта – 11.

Составьте программу, которая печатает таблицу умножения натуральных чисел в десятичной системе счисления.

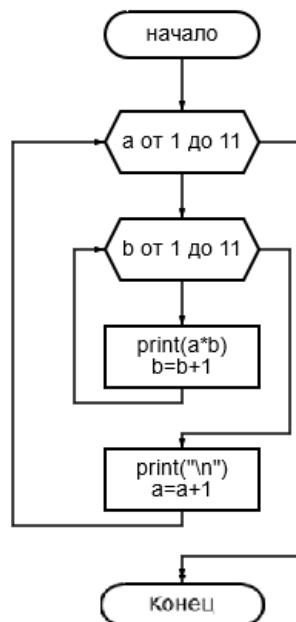


Рисунок 5 – UML-диаграмма задачи 3

```
for a in range(1, 11):
    for b in range(1,11):
        print(a*b, end = "\t")
    print("\n")
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Рисунок 6 – Решение задачи 3

Было выполнено задание повышенной сложности.

$$\text{Ci}(x) = \gamma + \ln x + \int_0^x \frac{\cos t - 1}{t} dt = \gamma + \ln x + \sum_{n=1}^{\infty} \frac{(-1)^n x^{2n}}{(2n)(2n)!}.$$

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math
import sys
# Постоянная Эйлера.
EULER = 0.5772156649015328606

# Точность вычислений.
EPS = 1e-10
if __name__ == '__main__':
    x = float(input("Value of x? "))
    if x == 0:
        print("Illegal value of x", file=sys.stderr)
        exit(1)
    a = x
    S, n = a, 1

    # Найти сумму членов ряда.
    while math.fabs(a) > EPS:
        a *= (((-1)**n)*x**(2*n)) / (2*n)*(2*n + 1)
        S += a
        n += 1

    # Вывести значение функции.
    print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")

Value of x? 0.3
Ei(0.3) = -0.3676668531405047
```

Рисунок 7 – Решение задачи повышенной сложности

**Вывод:** были приобретены навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры; освоены операторы языка Python версии 3.x if, while, for, break и continue, позволяющие

реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.