

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 1

з дисципліни «Об'єктно-орієнтоване програмування СУ»

Тема: «Розробка програм для математичних обчислень в Python»

ХАІ.301 .173. 310ст.1 ЛР

Виконав студент гр. \_\_\_\_\_310ст\_\_\_\_\_

\_\_\_\_\_Возвишаєв Олексій Андрійович\_\_\_\_\_

(підпис, дата)

(П.І.Б.)

Перевірів

\_\_\_\_\_к.т.н., доц. О. В. Гавриленко

\_\_\_\_\_ас. В. О. Білозерський

(підпис, дата)

(П.І.Б.)

## МЕТА РОБОТИ

Ознайомитися з основами розробки інтерпретованих програм і реалізувати скрипт для введення / виведення даних і виконання найпростіших математичних обчислень на мові програмування Python.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання з цілочисельними змінними. Всі вхідні і вихідні дані в задачах цієї групи є цілими числами. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число та ін.), вважаються додатними.

Завдання 2. Обчислити математичне вираз зі змінними дійсного типу, використовуючи стандартну бібліотеку `math`.

Завдання 3. Вирішити завдання з логічними змінними. У всіх завданнях даної групи потрібно вивести логічне значення `True`, якщо наведене висловлювання для запропонованих вихідних даних є істинним, і значення `False` в іншому випадку. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число і т.ін.), вважаються цілими додатними.

## ВИКОНАННЯ РОБОТИ

## Завдання 1 (Integer) 10

Вхідні дані: Дано тризначне число. Знайти суму і добуток його цифр.

num – тризначне число, введене користувачем. Тип: int. Діапазон допустимих значень: від 100 до 999 (включно).

hundreds – кількість сотень у тризначному числі. Тип: int. Діапазон: від 1 до 9.

tens – кількість десятків у тризначному числі. Тип: int. Діапазон: від 0 до 9.

ones – кількість одиниць у тризначному числі. Тип: int. Діапазон: від 0 до 9.

digit\_sum – сума цифр тризначного числа. Тип: int. Діапазон: від 1 до 27 (мінімум  $1 + 0 + 0$ , максимум  $9 + 9 + 9$ ).

digit\_product – добуток цифр тризначного числа. Тип: int. Діапазон: від 0 до 729 (мінімум  $0 * \text{будь-яка цифра}$ , максимум  $9 * 9 * 9$ ).

Вихідні дані:

num – тризначне число, яке вводить користувач.

Тип: int

Опис: Введене користувачем тризначне число, яке підлягає розбиттю на цифри.

hundreds – кількість сотень у тризначному числі.

Тип: int

Опис: Цифра, що відповідає сотням у тризначному числі. Вираховується діленням числа на 100.

digit\_product – добуток цифр тризначного числа.

Тип: int

Опис: Добуток цифр (сотень, десятків та одиниць) тризначного числа.

Відповідні текстові сповіщення:

"Enter a three-digit number:" – запит на введення тризначного числа.

"The number must be three digits!" – повідомлення про помилку, якщо введене число не є тризначним.

"Sum of digits: {digit\_sum}" – повідомлення з результатом суми цифр числа.

"Product of digits: {digit\_product}" – повідомлення з результатом добутку цифр числа.

Алгоритм вирішення показано нижче

Введення: Запросити тризначне число.

Перевірка: Якщо число не в діапазоні 100–999, вивести помилку і завершити програму.

Витягування цифр:

Сотні: `hundreds = num // 100`

Десятки: `tens = (num // 10) % 10`

Одиниці: `ones = num % 10`

Обчислення:

Сума цифр: `digit_sum = hundreds + tens + ones`

Добуток цифр: `digit_product = hundreds * tens * ones`

Вивід результату:

Лістинг коду вирішення задачі наведено в дод. А стор.9. Екран роботи програми показаний на рис. Б.1.

## Завдання 2. (math) 6

Вхідні дані:

$x$  — значення аргументу, яке вводиться користувачем для обчислення функції.

Тип: `float`

Обмеження:  $x > 7.5$  (оскільки логарифм не може бути визначений для значень  $x + 7.5 \leq 0$ )

Вихідні дані (ім'я, опис, тип):

$y$  — результат обчислення математичного виразу за заданим  $x$ .

Тип: `float`, якщо обчислення успішні.

`error_message` — текстове повідомлення, яке відображається у разі помилки (наприклад, недопустимі значення  $x$ , що можуть призвести до математичних помилок).

Тип: `str`, якщо виникає помилка.

Алгоритм вирішення показано нижче

Введення даних

Перевірка, чи  $x > -7.5x$ . Якщо умова не виконується, видається повідомлення про помилку.

Вирахування чисельника:

Обчислюється  $2x2^2$

Обчислюється  $|\sin(x)|$ .

Обчислюється  $|\tan(x)|$ .

Обчислюється  $2.5\cos^{(x)}$ .

Обчислюється сума і множення компонентів чисельника.

Підноситься до кореня 5-го ступеня.

Вирахування знаменника:

Обчислюється  $\log_2(x+7.5)$ , додається до 0.625.

Якщо  $x+7.5 \leq 0$ , видається повідомлення про помилку.

Обчислення виразу:

Ділимо чисельник на знаменник.

Виведення результату:

Виводиться результат  $y$ , або повідомлення про помилку, якщо є недопустимі значення.

Лістинг коду вирішення задачі наведено в дод. А стор.9. Екран роботи програми показаний на рис. Б.2.

Завдання 3 (Boolean) 13

Вхідні дані:

А — перше ціле число, введене користувачем.

Тип: int

Обмеження: будь-яке ціле число.

В — друге ціле число, введене користувачем.

Тип: int

Обмеження: будь-яке ціле число.

C — третє ціле число, введене користувачем.

Тип: int

Обмеження: будь-яке ціле число.

Вихідні дані:

res — результат перевірки істинності висловлювання «Хоча б одне з чисел A, B, C позитивне».

Тип: bool (значення True або False).

error\_message — текстове повідомлення про помилку у випадку неправильного введення (наприклад, якщо введено не ціле число).

Тип: str, у випадку виникнення помилки.

Алгоритм вирішення показано нижче

Введення даних: Користувач вводить три цілі числа A,B,C

У разі некоректного введення (не ціле число) виводиться повідомлення про помилку.

Перевірка істинності висловлювання: Виконується логічна операція: перевіряється, чи хоча б одне з чисел A,B,C є позитивним. Якщо умова виконується (хоча б одне число більше за 0), результатом є True, інакше — False.

Виведення результату: Виводиться результат перевірки істинності висловлювання.

Лістинг коду вирішення задачі наведено в дод. А стор.9. Екран роботи програми показаний на рис. Б.3.

## ВИСНОВКИ

Було вивчено методи перевірки умов за допомогою логічних операторів. На практиці закріплено навички введення та обробки цілих чисел, а також перевірки їх на позитивність. В коді програми відпрацьовано логіку перевірки істинності висловлювання та обробку помилок при введенні некоректних даних.

## ДОДАТОК А

### Лістинг коду програми до задач Завдання 1

```
def task_integer11():
    """Given a three-digit number, find the sum and product of its digits."""
    try:
        num = int(input("Enter a three-digit number: "))
        print(f"You entered: {num}") # Додаємо перевірку на введене число

        if num < 100 or num > 999:
            raise ValueError("The number must be three digits!")
    except ValueError as e:
        print(e)
        return # Зупиняємо виконання, якщо є помилка
    else:
        # Витягуємо цифри
        hundreds = num // 100
        tens = (num // 10) % 10
        ones = num % 10

        # Сума та добуток цифр
        digit_sum = hundreds + tens + ones
        digit_product = hundreds * tens * ones

        # Обчислюємо чисельник
        print(f"Sum of digits: {digit_sum}")
        print(f"Product of digits: {digit_product}")
```

### Лістинг коду програми до задач Завдання 2

```
import math

def calculate_y(x):
    try:
        # Обчислюємо чисельник
        numerator = (2 * x**2 - abs(math.sin(x)) * abs(math.tan(x)) *
                     2.5**math.cos(x))**(1/5)

        # Обчислюємо знаменник
        denominator = 0.625 + 2 * math.log2(x + 7.5)

        # Повний вираз
        y = numerator / denominator

    except ValueError:
        return "Помилка: неприпустимі значення для логарифму чи інших
        математичних операцій."
```

```
# Приклад використання
x = float(input("Введіть значення x: "))
result = calculate_y(x)
print(f"y = {result}")
```

### Лістинг коду програми до задач Завдання 3

```
def check_positive():
    try:
        # Ввод трёх целых чисел A, B, C
        A = int(input("Введите число A: "))
        B = int(input("Введите число B: "))
        C = int(input("Введите число C: "))
    except ValueError:
        print("Усі значення мають бути цілими числами!")
        return

    # Перевірка умови: хоча б одне із чисел позитивне
    res = A > 0 or B > 0 or C > 0

    # Висновок результату
    print( res)
```



## ДОДАТОК Б

### Скрін-шоти вікна виконання програми

```
def task_integer11():  
    """Given a three-digit number, find the sum and product of its digits."""  
    try:  
        num = int(input("Enter a three-digit number: "))  
        print(f"You entered: {num}") # Додаємо перевірку на введене число  
  
        if num < 100 or num > 999:  
            raise ValueError("The number must be three digits!")  
    except ValueError as e:  
        print(e)  
        return # Зупиняємо виконання, якщо є помилка  
    else:  
        # Витягуємо цифри  
        hundreds = num // 100  
        tens = (num // 10) % 10  
        ones = num % 10  
  
        # Сума та добуток цифр  
        digit_sum = hundreds + tens + ones  
        digit_product = hundreds * tens * ones  
  
        # Виводимо результат  
        print(f"Sum of digits: {digit_sum}")  
        print(f"Product of digits: {digit_product}")  
  
task_integer11()  
  
Enter a three-digit number: 123  
You entered: 123  
Sum of digits: 6  
Product of digits: 6  
task_integer11()  
  
Enter a three-digit number: 345  
You entered: 345  
Sum of digits: 12  
Product of digits: 60
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
Завдання1  
(Integer)

```
1 import math
2
3 def calculate_y(x):
4     try:
5         # Вычисляем числитель
6         numerator = (2 * x**2 - abs(math.sin(x)) * abs(math.tan(x)) * 2.5**math.cos(x))**(1/5)
7
8         # Вычисляем знаменатель
9         denominator = 0.625 + 2 * math.log2(x + 7.5)
10
11        # Полное выражение
12        y = numerator / denominator
13
14        return y
15    except ValueError:
16        return "Ошибка: недопустимые значения для логарифма или других математических операций."
17
18 # Пример использования
19 x = float(input("Введите значение x: "))
20 result = calculate_y(x)
21 print(f"y = {result}")
22
```

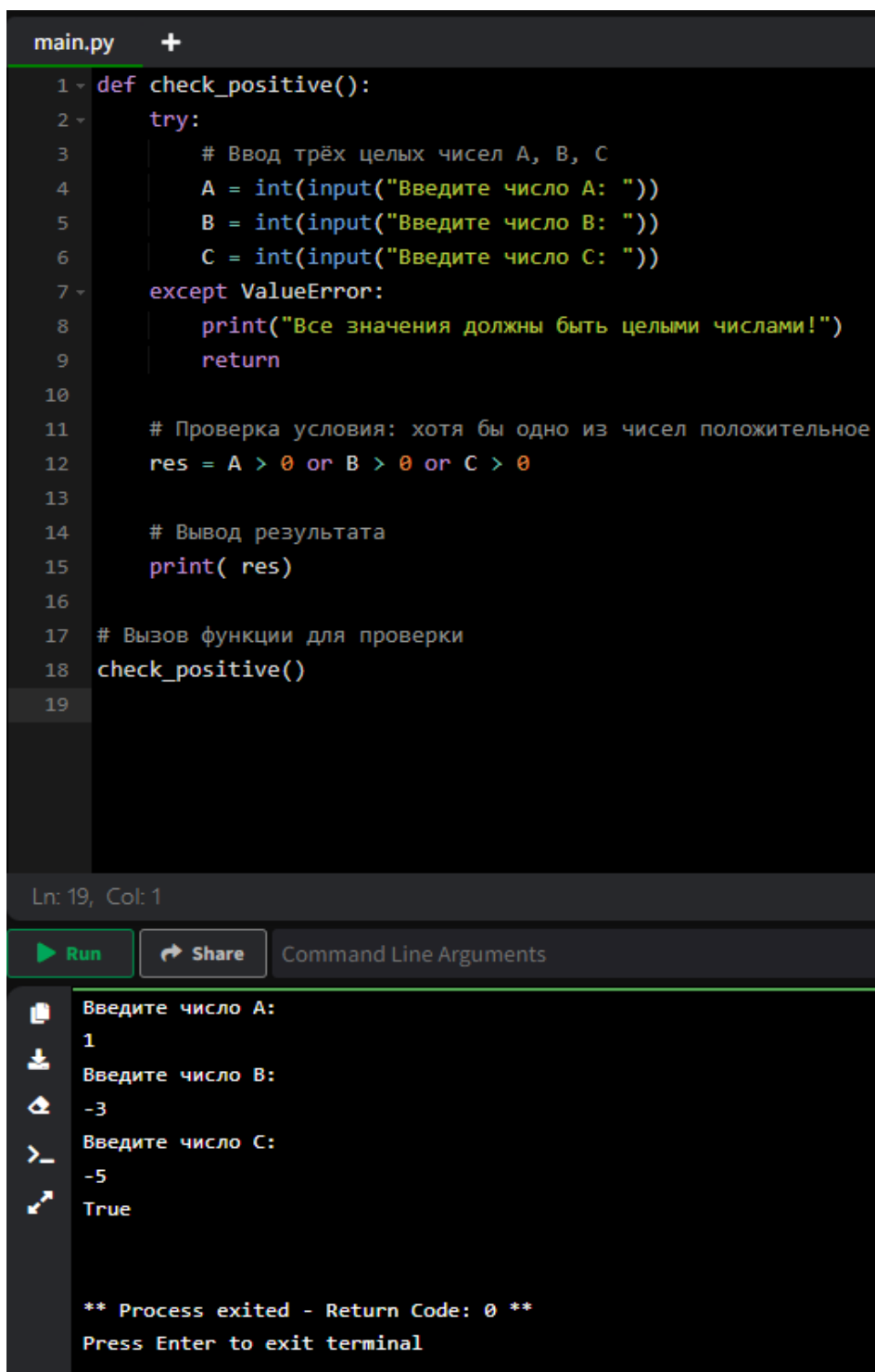
Ln: 17, Col: 1

**Run** **Share** Command Line Arguments

Введите значение x:  
2  
y = 0.20508852419648468

\*\* Process exited - Return Code: 0 \*\*  
Press Enter to exit terminal

Рисунок Б.2 – Экран виконання програми для вирішення завдання  
Завдання 2  
(math)



The image shows a Python IDE interface. At the top, a file named `main.py` is open. The code defines a function `check_positive()` that prompts the user for three integers A, B, and C. It uses a `try` block to handle potential `ValueError` exceptions. If the input is valid, it checks if at least one of the numbers is greater than zero (`A > 0 or B > 0 or C > 0`) and prints the result. The function is then called at the bottom of the script.

```
1 def check_positive():
2     try:
3         # Ввод трёх целых чисел A, B, C
4         A = int(input("Введите число A: "))
5         B = int(input("Введите число B: "))
6         C = int(input("Введите число C: "))
7     except ValueError:
8         print("Все значения должны быть целыми числами!")
9         return
10
11     # Проверка условия: хотя бы одно из чисел положительное
12     res = A > 0 or B > 0 or C > 0
13
14     # Вывод результата
15     print( res)
16
17 # Вызов функции для проверки
18 check_positive()
19
```

Below the code editor, the status bar shows "Ln: 19, Col: 1". There are buttons for "Run" (a green play icon) and "Share" (a share icon). To the right of these buttons is a field for "Command Line Arguments".

The output of the program is shown in a terminal window below the editor. It displays the prompts and user input for A, B, and C, followed by the boolean result `True`. At the bottom, it shows the process exit message: `** Process exited - Return Code: 0 **` and a prompt to "Press Enter to exit terminal".

```
Введите число A:
1
Введите число B:
-3
Введите число C:
-5
True

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Рисунок Б.3 – Экран виконання програми для вирішення завдання  
Завдання 3  
(Boolean)