Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Програмування. Частина 2.

Лабораторна робота №1

«Основні типи та оператори мови програмування Java (Python)»

Виконав:
студент гр. ІО-41
Давидчук А. М.
Залікова книжка № 4106

Перевірив
Коренко Д.В.

Київ-2025

Тема: «Основні типи та оператори мови програмування Java (Python)».

Мета: Ознайомлення з основними типами та операторами в Java (Python). Здобуття навичок у використанні типів та операторів в Java (Python).

Порядок роботи

Через те, що є можливість використовувати мову Python, для виконання лабораторних робіт з ООП, то я надалі буду його використовувати як основний засіб.

Моя залікова книжка 4106, значить:

$C_2 = 4106 \bmod 2 = 0$
$C_3 = 4106 \bmod 3 = 2$
$C_5 = 4106 \bmod 5 = 1$
$C_7 = 4106 \bmod 7 = 4$

Згідно з таблицями варіантів, мій варіант:

C_2	Операція O_1
0	+
1	-

C_5	Операція O_2
0	*
1	/
2	%
3	+
4	-

C_7	тип індексів *i* та *j*
0	byte
1	short
2	int
3	long
4	char
5	float
6	double

А константа $C = C_3 = 2$.

Значить фінальна формула буде:

$$S = \sum_{i=a}^{n} \sum_{j=b}^{m} \frac{i / j}{i + 2}$$

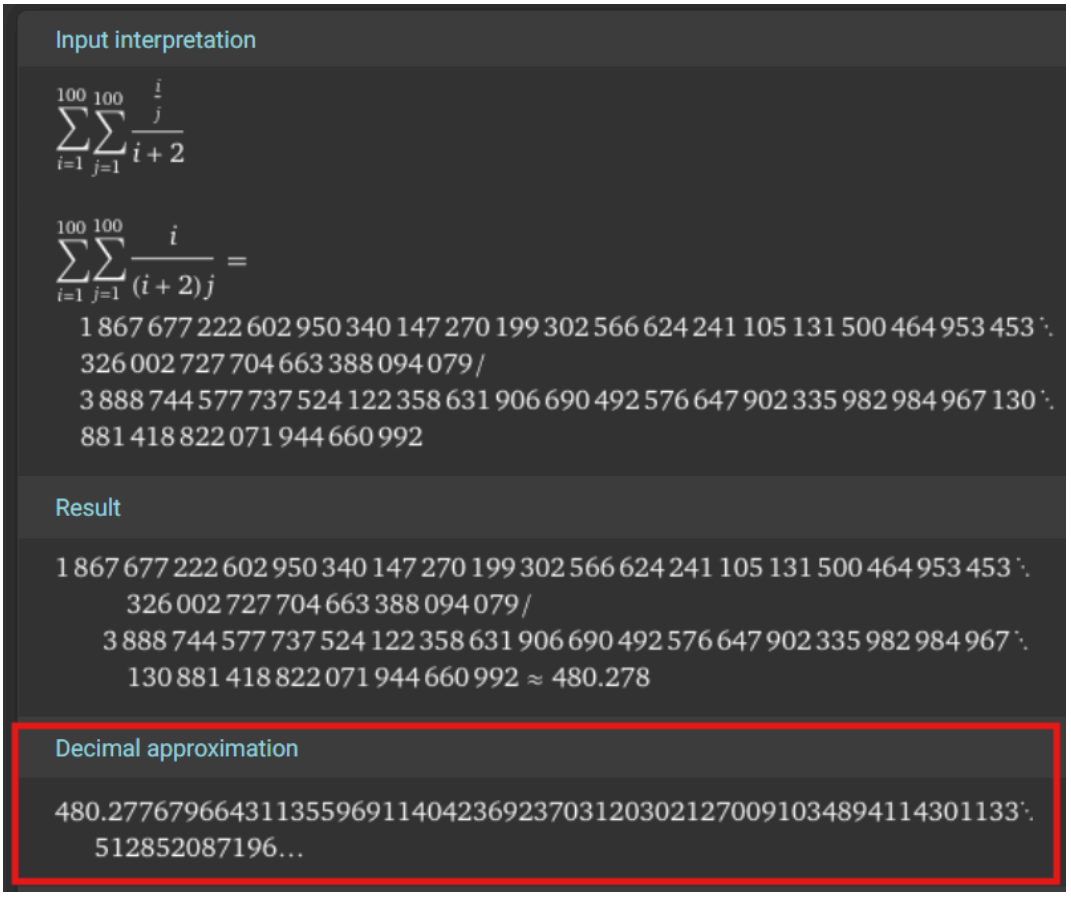
Код (Python):

```
class char16bit:
    # Через те, що в мові Java тип char являє собою unsigned 16bit ціле число
    # значить значення мають бути в діапазоні [0; 65536]
    # Я вручну напишу клас, який матиме схожі властивості

    def __init__(self, value):

        # Логіка при переповнених значеннях з діапазону [0; 65536] -> 16bit
число зі знаком:
        if value < 0 or value > 65536: value % 65536

        self.value = value

    def __le__(self, other):

        # Використовується при операції <=
        return self.value <= other

    def __add__(self, other):

        # Використовується при операції +
        return char16bit(self.value + other)

    def __truediv__(self, other):

        # Використовується при операції /
        return char16bit(self.value / other.value)

    def __float__(self):

        # Використовується при виведенні значення до float
        return self.value

    def __str__(self):

        # Реалізовано, для відповідності типу
        return chr(self.value)

class Sum:

    def __init__(self, a, b, n, m):
```

```
        # Перевіряємо, щоб нижні границі суми не були більшими за верхні:

        if a > n or b > m: if a > n or b > m: print(f"Wrong input limits of
the summation"); exit()
        self.a = a
        self.b = b
        self.n = n
        self.m = m

    def calculate_and_print(self):

        result = 0

        # Перетворюємо в конкретний числовий тип для обчислення
        # Типи індексів завжди мають бути типом char16bit:

        i = char16bit(self.a)
        j = char16bit(self.b)

        # Проводимо обчислення:

        while i <= self.n:

            while j <= self.m:

                try:

                    result += float((i / j) / (i + 2))

                except ZeroDivisionError:

                    # В разі ділення на нуль
                    return f"ZeroDivivsionError"

                j += 1

            i += 1
            j = char16bit(self.b)

        return result

sum_result = Sum(int(input("a: ")), int(input("b: ")), int(input("n: ")),
int(input("m: "))) # Введення значень a, b, n, m
print(sum_result.calculate_and_print()) # Виводимо результат
```

Результати виконання та перевірка значень:

```
PS C:\Users\artem\OneDrive\Desktop\Important folder\University\OOP\Lab1> py lab1.py
a: 1
b: 1
n: 100
m: 100
480.2776796643114
```




```
PS C:\Users\artem\OneDrive\Desktop\Important folder\University\OOP\Lab1> py lab1.py
a: -1
b: -1
n: 100
m: 100
ZeroDivivsionError
```

```
PS C:\Users\artem\OneDrive\Desktop\Important folder\University\OOP\Lab1> py lab1.py
a: -1
b: -1
n: -100
m: -100
Wrong input limits of the summation
```

Висновок:

У ході виконання лабораторної роботи було реалізовано клас char16bit, що імітує поведінку 16-бітного беззнакового типу char з Java. Основними особливостями класу є контроль діапазону значень [0; 65535] та підтримка базових арифметичних операцій. Також було розроблено клас Sum, який обчислює суму виразу згідно з заданими межами індексів. Реалізація використовує char16bit для ітерації, що забезпечує коректну обробку значень у межах 16-бітного діапазону. Програма враховує можливість ділення на нуль, обробляючи такі випадки за допомогою ZeroDivisionError та ввід некоректних значень границь суми. Таким чином, виконана робота демонструє принципи роботи з користувацькими типами даних та реалізацію математичних обчислень у Python.