

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт
з лабораторної роботи № 4 з
дисципліни «Алгоритми та структури
даних-1. Основи алгоритмізації»
«Дослідження арифметичних
циклічних алгоритмів»
Варіант 2

Виконав студент ПІ-12, Басараб Олег Андрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів _____
(прізвище, ім'я, по батькові)

Лабораторна робота №4 “Дослідження арифметичних циклічних алгоритмів”

Варіант 2

Мета – дослідити особливості роботи арифметичних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

Задача 2. Дано натуральне число n . Обчислити

$$\sum_{k=1}^n \frac{1}{k(2k+1)^2}$$

Розв'язок

Постановка задачі. Результатом розв'язку є дійсне число `seriesSum`. Для знаходження `seriesSum` повинне бути задане ціле число $n \geq 1$. Інших початкових даних для розв'язку задачі не потрібно.

Побудова математичної моделі. Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Кількість елементів ряду, суму яких потрібно знайти	Цілий	n	Вхідне дане
Порядковий номер елементу ряду	Цілий	k	Проміжне дане
Поточний елемент ряду	Дійсний	currentElem	Проміжне дане
Сума елементів ряду з 1-го по n-ий	Дійсний	seriesSum	Результат

Таким чином, формулювання завдання зводиться до:

- 1) задання початкових значень змінних, а саме `seriesSum = 0`;
- 2) знаходження суми елементів ряду з 1-го по n-ий за допомогою арифметичного циклу. Лічильником арифметичного циклу є змінна `k` з

початковим значенням 1 та кінцевим значенням n. Вважаємо, що умовою невиходу з циклу є $k < n + 1$. Крок дорівнює 1. В тілі циклу спершу знаходимо поточний елемент за формулою $\text{currentElem} = 1.0 / (k * (2 * k + 1) * (2 * k + 1))$. Зауважимо, що в формулі використано запис одиниці «1.0», щоб зазначити дійсний тип змінної currentElem. Наступною дією додаємо поточний елемент ряду до суми елементів ряду $\text{seriesSum} += \text{currentElem}$.

Розіб'ємо алгоритм на кроки:

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію задання початкових значень змінних.

Крок 3. Деталізуємо дію знаходження суми елементів ряду з 1-го по n-ий.

Програмні специфікації запишемо у псевдокоді та в графічній формі в блок-схемі алгоритму.

Псевдокод.

Крок 1

початок

введення a

задання початкових значень
змінних

знаходження суми елементів ряду
з 1-го по n-ий

вивід seriesSum

кінець

Крок 2

початок

введення a

seriesSum = 0

знаходження суми елементів ряду
з 1-го по n-ий

вивід seriesSum

кінець

Крок 3

початок

введення a

$seriesSum = 0$

повторити

для k від 1 до $n + 1$

$currentElem = 1.0 / (k * (2 * k + 1) * (2 * k + 1))$

$seriesSum += currentElem$

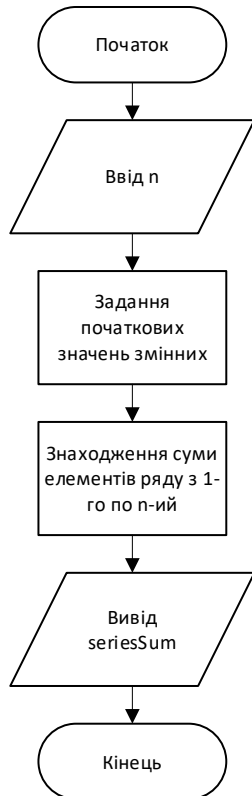
все повторити

вивід $seriesSum$

кінець

Блок-схема алгоритму.

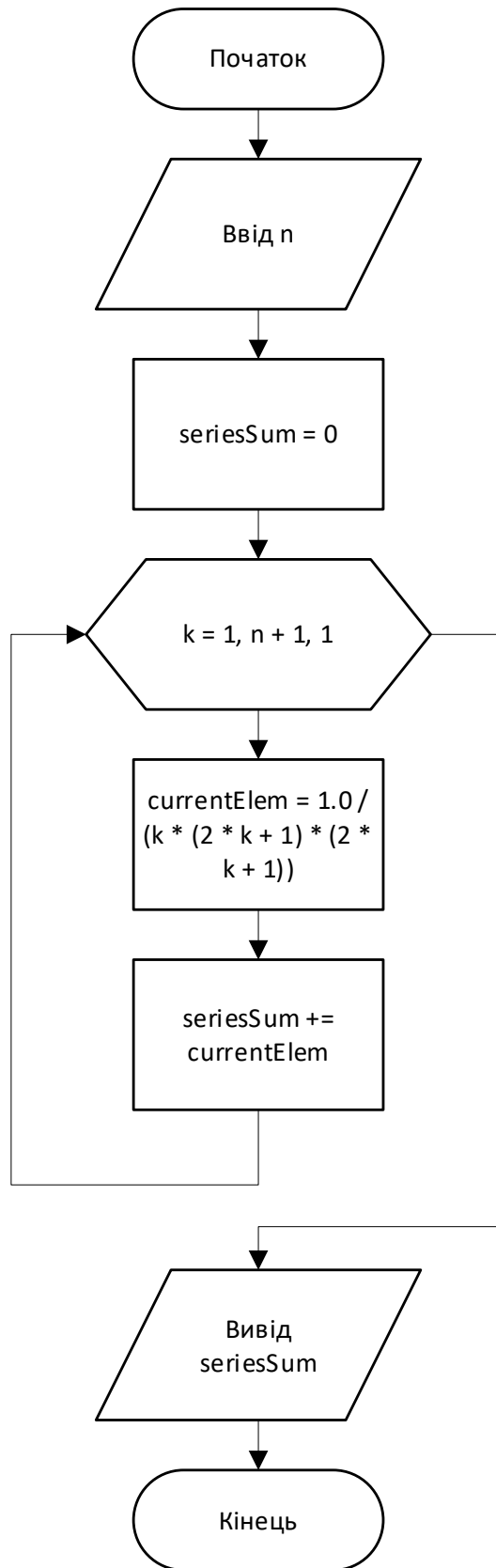
Крок 1



Крок 2



Крок 3



Випробування алгоритму.

Перевірка №1

Блок	Дія
	Початок
1	Введення: 5
2	seriesSum = 0
3	повторити для k від 1 до n + 1 currentElem = $1.0 / (k * (2 * k + 1) * (2 * k + 1))$ seriesSum += currentElem все повторити
4	Вивід: 0.142653
	Кінець

На скріншоті можна побачити усі елементи ряду, отримані в результаті виконання алгоритму з такими вхідними даними:

```
Element[1] = 0.111111; Element[2] = 0.02; Element[3] = 0.00680272; Element[4] = 0.00308642; Element[5] = 0.00165289;
```

Перевірка №2

Блок	Дія
	Початок
1	Введення: 3
2	seriesSum = 0

3	<p>повторити</p> <p>для k від 1 до n + 1</p> <p>currentElem = 1.0 / (k * (2*k + 1) * (2 * k + 1))</p> <p>seriesSum += currentElem</p> <p>все повторити</p>
4	Вивід: 0.137914
	Кінець

На скріншоті можна побачити усі елементи ряду, отримані в результаті виконання алгоритму з такими вхідними даними:

```
Element[1] = 0.111111; Element[2] = 0.02; Element[3] = 0.00680272;
```

Висновки. Таким чином, в результаті виконання лабораторної роботи було досліджено особливості роботи арифметичних циклів на прикладі побудови циклу для обчислення членів ряду з 1-го по n-ий та знаходження їх суми та набуто практичних навичок їх використання під час складання програмних специфікацій.