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 № 3 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження ітераційних циклічних алгоритмів»

Варіант 22

Виконав студент ІП-12 Мельник Михайло Олександрович

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 3**

**Дослідження ітераційних циклічних алгоритмів**

**Мета** – дослідити подання операторів повторення дій та набути практичних навичок їх використання під час складання циклічних програмних специфікацій.

**Варіант 22**

**Розв’язання:**

1. **Постановка задачі**

Результатом розв’язку задачі є число Ейлера, обчисленого як суму ряду із заданою точністю. Для визначення результату достатньо задати точність E. Інших початкових значень для розв’язку не потрібно.

1. **Побудова математичної моделі**

***Складемо таблицю імен змінних***

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Точність обчислень | Дійсний | Е | Початкове дане |
| Сума ряду | Дійсний | sum | Результат |
| Член ряду | Дійсний | term | Проміжне дане |
| Порядковий номер члена рялу | Цілий | i | Проміжне дане |

Таким чином, математичне формулювання завдання зводиться до обчислення суми ряду поки кожен наступний член більший за E (оскільки term→0), і де кожен член ряду дорівнює попередньому членові, поділеному на порядковий номер даного: term = term / i. Оскільки кожен член прогресії ∈ (0, 1], зробимо перевірку на цю умову.

1. **Розроблення алгоритму**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію перевірки введеного Е

*Крок 3.* Деталізуємо дію обчислення значень початкового члена, його номера та суми ряду

*Крок 4.* Деталізуємо дію обчислення числа Ейлера із заданою точністю

***Псевдокод***

*крок 1*

**початок**

**ввід** E

перевірка введеного E

обчислення значень початкового

члена, його номера та суми ряду

обчислення числа Ейлера

із заданою точністю

**вивід** sum

**кінець**

*крок 2*

**початок**

**ввід** E

**якщо** E <= 0 || E > 1

**то**

**вивід «**Invalid precision»

**інакше**

обчислення значень початкового

члена, його номера та суми ряду

обчислення числа Ейлера

із заданою точністю

**все якщо**

**вивід** sum

**кінець**

*крок 3*

**початок**

**ввід** E

**якщо** E <= 0 || E > 1

**то**

**вивід «**Invalid precision»

**інакше**

term = 1.0

i = 1

sum = 0.0

обчислення числа Ейлера

із заданою точністю

**все якщо**

**вивід** sum

**кінець**

*крок 4*

**початок**

**ввід** E

**якщо** E <= 0 || E > 1

**то**

**вивід «**Invalid precision»

**інакше**

term = 1.0

i = 1

sum = 0.0

**поки** term >= E

**повторити**

sum := sum + term

term := term / i

i := i + 1

**все повторити**

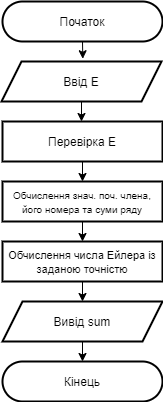
**все якщо**

**вивід** sum

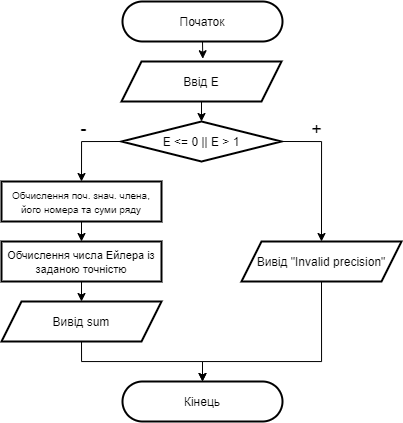
**кінець**

***Блок-схема***

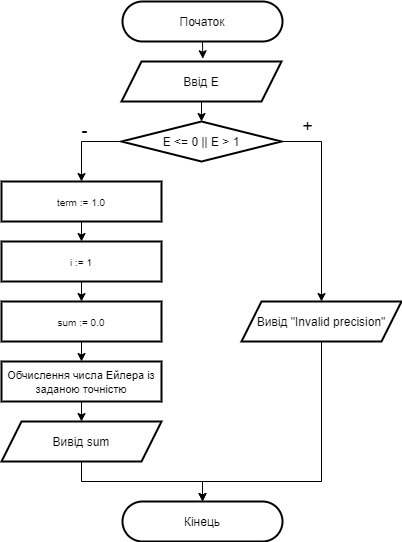
*Крок 1*

**

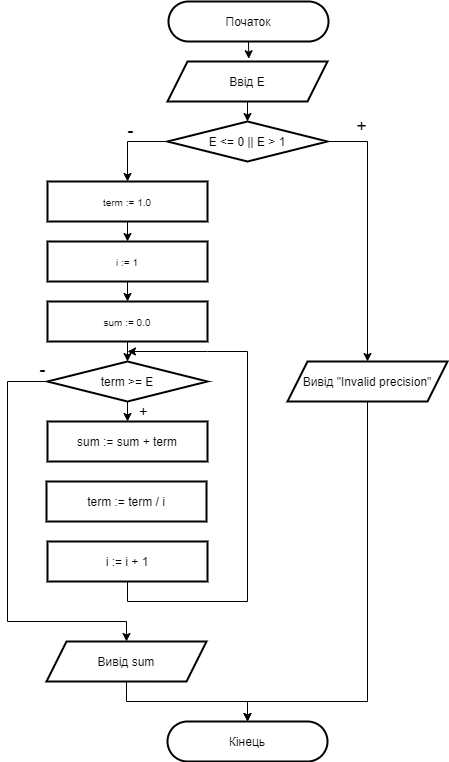
*Крок 2*

**

*Крок 3*

**

*Крок 4*

**

1. **Випробування алгоритму**

Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних: E=0, E=1, E=0.4

|  |  |  |  |
| --- | --- | --- | --- |
| **Блок** | **Дія** | **Блок** | **Дія** |
|  | Початок |  | Початок |
| 1 | Ввід 0 | 1 | Ввід 1 |
| 2 | 0 <= 0 || 0 > 1 істина | 2 | 1 <= 0 || 1 > 1 хибне |
|  |  | 3 | sum := 0, term :=1, i := 1, |
| 6 | Invalid precision | 4 | 1 >= 1 істина |
|  |  | 3 | sum := 1, term := 1, i :=2 |
|  |  | 4 | 1 >= 1 істина |
|  |  | 3 | sum := 2, term := 1/2, i :=3 |
|  |  | 4 | 1/2 >= 1 хибне |
|  |  | 5 | вивід 2 |
|  | Кінець |  | Кінець |

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | Початок |
| 1 | Ввід 0.4 |
| 2 | 0.4 <= 0 || 0.4 > 1 хибне |
| 3 | sum := 0, term :=1, i := 1 |
| 4 | 1 >= 0.4 істина |
| 3 | sum := 1, term := 1, i :=2 |
| 4 | 1 >= 0.4 істина |
| 3 | sum := 2, term := 1/2, i :=3 |
| 4 | 1/2 >= 0.4 істина |
| 3 | sum := 2.5, term := 1/6, i :=4 |
| 4 | 1/6 >= 0.4 хибне |
| 5 | Вивід 2.5 |
|  | Кінець |

Алгоритм відповідає поставленій задачі, помилок не виявлено.

1. **Висновки**

Під час виконання даної лабораторної роботи було розроблено й випробувано алгоритм обчислення числа Ейлера із заданою точністю, досліджено подання операторів повторення дій та набуто практичних навичок їх використання під час складання циклічних програмних специфікацій.