

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

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 5

з дисципліни «Алгоритмізація та програмування»
на тему "Реалізація циклічних алгоритмів мовою C ++"

XAI.301. 174. 319. 5ЛР

Виконав студент гр. 319

Женя Качка

(підпис, дата)

(П.І.Б.)

Перевірів

 К.Т.Н.,

доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису мовою C ++ і поданням у вигляді UML діаграм циклічних алгоритмів і реалізувати алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованого циклу мовою C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 4 геометричні області:Реалізувати меню в командному вікні, яке дозволяє багаторазово виконувати завдання та перевіряти вхідні дані на коректність.

Завдання 23 підрахунок рядів:Обчислити заданий ряд і перевірити його збіжність.

Завдання 18 Збіжність/розбіжність рядів:Вивести всі елементи ряду, які є кратними заданому числу

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі 4 геометричні області

Вхідні дані (ім'я, опис, тип, обмеження):

Номер завдання: 4, 18, 23, -1.

Вихідні дані (ім'я, опис, тип):

Результат виконання відповідного завдання (або повідомлення про вихід).

Алгоритм вирішення показано на малюнку .1

малюнок 1.

Малюнок 1 – задача 4 геометричні області

```
✓ #include <iostream>
  #include <cmath>
  #include <locale> // Для використання setlocale

  using namespace std;

  // Функція для задачі 4
  ✓ double task4(int terms) {
    double sum = 0.0;
    ✓ for (int k = 1; k <= terms; k++) {
      sum += pow(k, k) / (k + 1);
    }
    return sum;
  }
```

Лістинг коду вирішення задач 4 геометричні області,Завдання 23 підрахунок рядів

наведено в дод. А (стор. 7).

Екран роботи програми показаний на рис. Б.8.

Завдання 2.

Вирішення задачі 23 підрахунок рядів

Вхідні дані (ім'я, опис, тип, обмеження):

Значення x: реальне число.

Вихідні дані (ім'я, опис, тип):

Значення суми ряду до умови збіжності.

Алгоритм вирішення показано на малюнку .2

Малюнок 2 – задача 23 підрахунок рядів

```

// Функция для задачи 23
double task23(double x, int terms) {
    double sum = 0.0;
    for (int n = 1; n <= terms; n++) {
        double numerator = pow(x, 3 * n); // x^(3n)
        long long factorial = 1;
        for (int i = 2; i <= 2 * n + 1; i++) {
            factorial *= i; // Вычисляем (2n+1)!
        }
        sum += numerator / factorial; // Добавляем к сумме
    }
    return sum;
}

```

Завдання 3.

Вирішення задачі 18 Збіжність/розбіжність рядів

Вхідні дані (ім'я, опис, тип, обмеження):

Кількість елементів n: ціле число, кратність k: ціле число.

Вихідні дані (ім'я, опис, тип):

Елементи ряду, що є кратними числу k.

Алгоритм вирішення показано на малюнку .3

Малюнок 3 – задача 18 Збіжність/розбіжність рядів

```

16 // Функция для задачи 23
17 double task23(double x, int terms) {
18     double sum = 0.0;
19     for (int n = 1; n <= terms; n++) {
20         double numerator = pow(x, 3 * n); //  $x^{(3n)}$ 
21         long long factorial = 1;
22         for (int i = 2; i <= 2 * n + 1; i++) {
23             factorial *= i; // Вычисляем  $(2n+1)!$ 
24         }
25         sum += numerator / factorial; // Добавляем к сумме
26     }
27     return sum;
28 }
29
30 // Функция для задачи 18 (примерная формула)
31 double task18(int terms) {
32     double sum = 0.0;
33     for (int n = 1; n <= terms; n++) {
34         sum += (n * n) / pow(2, n); // Пример:  $n^2/2^n$ 
35     }
36     return sum;
37 }
38
39 int main() {
40     setlocale(LC_ALL, "RU"); // Устанавливаем русский язык для вывода
41     int choice;
42     cout << "Выберите задачу (1 - задача 4, 2 - задача 23, 3 - задача 18): ";
43     cin >> choice;
44
45     int terms;
46     double x;
47
48     switch (choice) {
49     case 1:
50         cout << "Введите количество членов для задачи 4: ";
51         cin >> terms;
52         cout << "Сумма для задачи 4: " << task4(terms) << endl;
53         break;
54
55     case 2:
56         cout << "Введите значение x и количество членов для задачи 23: ";
57         cin >> x >> terms;
58         cout << "Сумма для задачи 23: " << task23(x, terms) << endl;
59         break;
60
61     case 3:
62         cout << "Введите количество членов для задачи 18: ";
63         cin >> terms;
64         cout << "Сумма для задачи 18: " << task18(terms) << endl;
65         break;
66
67     default:
68         cout << "Некорректный выбор задачи." << endl;
69     }
70
71     return 0;
72 }

```

ВИСНОВКИ

Під час виконання лабораторної роботи №5 були реалізовані та протестовані циклічні алгоритми мовою програмування C++.

У процесі роботи:

1) Створено зручне меню для вибору завдань, що дозволяє багаторазово виконувати їх без перезапуску програми.

2) Реалізовано обчислення ряду з перевіркою збіжності, що продемонструвало практичне застосування умов закінчення циклу

3) Виконано завдання з визначення елементів ряду, кратних заданому числу, що дозволило закріпити навички роботи з умовами в циклах.

Робота підтвердила важливість використання циклів для розв'язання типових задач алгоритмізації, а також підвищила розуміння структури програм мовою C++.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#include <cmath>
#include <locale> // Для использования setlocale

using namespace std;

// Функция для задачи 4
double task4(int terms) {
    double sum = 0.0;
    for (int k = 1; k <= terms; k++) {
        sum += pow(k, k) / (k + 1);
    }
    return sum;
}

// Функция для задачи 23
double task23(double x, int terms) {
    double sum = 0.0;
    for (int n = 1; n <= terms; n++) {
        double numerator = pow(x, 3 * n); //  $x^{(3n)}$ 
        long long factorial = 1;
        for (int i = 2; i <= 2 * n + 1; i++) {
            factorial *= i; // Вычисляем  $(2n+1)!$ 
        }
        sum += numerator / factorial; // Добавляем к сумме
    }
    return sum;
}

// Функция для задачи 18 (примерная формула)
double task18(int terms) {
    double sum = 0.0;
    for (int n = 1; n <= terms; n++) {
        sum += (n * n) / pow(2, n); // Пример:  $n^2/2^n$ 
    }
    return sum;
}

int main() {
    setlocale(LC_ALL, "RU"); // Устанавливаем русский язык для вывода
    int choice;
    cout << "Выберите задачу (1 - задача 4, 2 - задача 23, 3 - задача 18): ";
    cin >> choice;

    int terms;
    double x;

    switch (choice) {
        case 1:
            cout << "Введите количество членов для задачи 4: ";
            cin >> terms;
            cout << "Сумма для задачи 4: " << task4(terms) << endl;
            break;

        case 2:
            cout << "Введите значение x и количество членов для задачи 23: ";
            cin >> x >> terms;
```

```
    cout << "Сумма для задачи 23: " << task23(x, terms) << endl;
    break;

case 3:
    cout << "Введите количество членов для задачи 18: ";
    cin >> terms;
    cout << "Сумма для задачи 18: " << task18(terms) << endl;
    break;

default:
    cout << "Некорректный выбор задачи." << endl;
}

return 0;
}
```


ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
Выберите задачу (1 - задача 4, 2 - задача 23, 3 - задача 18): 1
Введите количество членов для задачи 4: 4
Сумма для задачи 4: 59.7833

C:\Users\tvink\source\repos\практическая номер 5\x64\Debug\практическая номер 5.exe (процесс 12744) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання 4 геометричні області

```
Выберите задачу (1 - задача 4, 2 - задача 23, 3 - задача 18): 2
Введите значение x и количество членов для задачи 23: 3
4
Сумма для задачи 23: 15.9449

C:\Users\tvink\source\repos\практическая номер 5\x64\Debug\практическая номер 5.exe (процесс 15164) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок Б.2 – Экран виконання програми для вирішення завдання 23 підрахунок рядів

```
Выберите задачу (1 - задача 4, 2 - задача 23, 3 - задача 18): 3
Введите количество членов для задачи 18: 2
Сумма для задачи 18: 1.5

C:\Users\tvink\source\repos\практическая номер 5\x64\Debug\практическая номер 5.exe (процесс 12388) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок Б.3 – Экран виконання програми для вирішення завдання 18 Збіжність/розбіжність рядів