

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
Національний технічний університет України  
Факультет інформатики та обчислювальної техніки  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Кафедра інформаційних систем та технологій

Звіт  
з лабораторної роботи № 5  
**«Реалізація алгоритмів з використанням функцій та показчиків»**  
з дисципліни  
**«Програмування – 1. Основи програмування»**

**Варіант № 25**

**Перевірів:**

**доц. Корнага Ярослав Ігорович**

**Виконала: Павлова Софія**

**Студентка гр. ІС-12 , ФІОТ**

**1 курс,**

**залікова книжка № ІС-1224**

Київ 2021

## **Лабораторна робота № 5**

**Тема:** Реалізація алгоритмів з використанням функцій та покажчиків.

**Мета:** Ознайомитися з особливостями посилальних типів даних, опанувати технологію застосування посилальних типів даних, навчитися розробляти алгоритми та програми із застосуванням посилальних типів даних.

**Обладнання:** Персональні комп'ютери.

## ЗАВДАННЯ

### Завдання №1

Обчислити значення функції  $y$ , розвинувши функцію  $\cos(x)$  у ряд Тейлора. Аргумент  $x$  змінюється від -2 до 2 з кроком 0.5. Визначити

1. похибку.

$$y = \begin{cases} \cos(x/2) / \cos(x^2), & -1 \leq x \leq 0, \\ \cos^2(x/2) * \cos(2 * x), & x > 0. \end{cases}$$

Ряд Тейлора для  $\cos(x)$ :

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^{n-1} x^{2n-2}}{(2n-2)!}$$

#### Алгоритм:

1. Увести початкове, кінцеве значення аргументу функції та його крок зміни.
2. Повторювати дії:
  - 2.1. Задати поточне значення аргументу функції, що дорівнює початковому значенню;
  - 2.2. Якщо поточне значення аргументу не більше за кінцеве, то виконати такі дії:
    - 2.2.1. Якщо аргумент функції знаходиться в межах від -1 до 2, то визначити значення функції за стандартною функцією та за формулою ряду Тейлора, виконавши дії:
      - 2.2.1.1. Визначити початкову суму ряду, що дорівнює 1;
      - 2.2.1.2. Задати точність розрахунків поточного елемента ряду;
      - 2.2.1.3. Задати початкову кількість елементів ряду, що дорівнює 1.
      - 2.2.1.4. Поки значення поточного елемента ряду більше за значення точності розрахунків повторювати такі дії:
        - визначити значення елемента ряду;
        - додати значення елемента ряду до суми;
        - перейти до наступного елемента ряду.
    - 2.2.2. Визначити похибку.
    - 2.2.3. Якщо аргумент функції менший за -1, то визначити значення функції за формулою ряду Тейлора, стандартною функцією та похибку.

2.2.4. Якщо функцію не визначено при заданих значеннях аргументу, то задати значення 1 для ознаки невизначеності функції, вивести відповідне повідомлення, інакше вивести значення функції та похибку.

2.3. Змінити поточне значення аргументу функції.

2.4. Перейти до дій п. 2.2.

3. Кінець повторень

Код:

```
#include <iostream>
#include <math.h>
using namespace std;

float accuracy;

//===== Функция развития cos в ряд Тейлора =====//
float cos_Taylor(float a) {
    float sum = 1;
    float n = 1;
    float an = 1;
    while (an > accuracy) {
        an *= pow((-1), (n - 1)) * pow(a, 2) / (n * (n + 1));
        sum += an;
        n++;
    }
    return sum;
}

int main() {
    setlocale(LC_ALL, "Rus");
    double x, x1, xn, step, accuracy, y = 0, y_Taylor = 0, mistake = 0;

    cout << "-----\n";
    cout << "Функция определена для аргумента X > -1, меняющегося с шагом 0.5\n\n";
    cout << "  cos(x/2) / cos(x^2)  ,  -1<=x<=0\n";
    cout << "y = \n";
    cout << "  cos^2(x/2) * cos(2x)  ,  x>0\n";
    cout << "-----\n\n";

    cout << "Введите начальное значение функции : ";
    cin >> x1;
    if ((x1 - int(x1)) != 0) {
        cout << "\n===== Error
=====^n";
```

```

    cout << " Начальное значение НЕ удовлетворяет условие\n";
    exit(0);
}

cout << "Введите конечное значение функции : ";
cin >> xn;
if (xn < x1 || (xn - int(xn)) != 0) {
    cout << "\n/===== Error
=====^n";
    cout << " Конечное значение НЕ удовлетворяет условие\n";
    exit(0);
}

cout << "Введите шаг функции : ";
cin >> step;
if (step > (xn - x1)) {
    cout << "\n/===== Error
=====^n";
    cout << " Шаг НЕ удовлетворяет условие\n";
    exit(0);
}

cout << "Введите точность расчётов для ряда Тейлора : ";
cin >> accuracy;

for (x = x1; x <= xn; x += step) {
    if (x >= -1 && x <= 0) {
        y = cos(x / 2) / cos(pow(x, 2));
        y_Taylor = cos_Taylor(x / 2) / cos_Taylor(pow(x, 2));
        mistake = y - y_Taylor;
    }
    else if (x > 0) {
        y = pow(cos(x / 2), 2) * cos(2*x);
        y_Taylor = pow(cos_Taylor(x / 2), 2) * cos_Taylor(2 * x);
        mistake = y - y_Taylor;
    }
    else {
        cout << "\n/===== Error
=====^n";
        cout << " При X = " << x << ", функция не определена\n";
        continue;
    }
    cout << "\n-----^n";
    cout << "При X = " << x << ":\nФункция = " << y << "\nТейлор = " << y_Taylor <<
"\nПогрешность = " << mistake << "\n";
}
}

```

Результат:

```
-----
Функция определена для аргумента X > -1, меняющегося с шагом 0.5
      cos(x/2) / cos(x^2)   ,   -1<=x<=0
y =
      cos^2(x/2) * cos(2x)  ,   x>0
-----

Введите начальное значение функции : -2
Введите конечное значение функции : 2
Введите шаг функции : 0.5
Введите точность расчётов для ряда Тэйлора : 15

/===== Error =====/
При X = -2, функция не определена

/===== Error =====/
При X = -1.5, функция не определена

-----
При X = -1:
Функция = 1.62424
Тэйлор = 1.62424
Погрешность = -2.22045e-16

-----
При X = -0.5:
Функция = 1
Тэйлор = 1
Погрешность = 0

-----
При X = 0:
Функция = 1
Тэйлор = 1
Погрешность = 0

-----
При X = 0.5:
Функция = 0.507231
Тэйлор = 0.507231
Погрешность = 1.11022e-16

-----
При X = 1:
Функция = -0.320496
Тэйлор = -0.320496
Погрешность = 5.55112e-17

-----
При X = 1.5:
Функция = -0.530011
Тэйлор = -0.530011
Погрешность = 0

-----
При X = 2:
Функция = -0.190816
Тэйлор = -0.190816
Погрешность = -1.11022e-16
```

Рис. 1. Выведений результат виконання завдання 1

## Завдання №2

1. Надрукувати таблиці значень функцій  $\cos(x)$  та  $\sin(x)$  на відрізку  $[a; b]$  із кроком  $h$ , розвинувши функції у ряд Тейлора та використавши покажчики на функції.

### Алгоритм:

1. Увести початкове, кінцеве значення аргументу функції та його крок зміни.
2. Повторювати дії:
  - 2.1. Задати поточне значення аргументу функції, що дорівнює початковому значенню;
  - 2.2. Визначити початкову суму ряду, що дорівнює 1;
  - 2.3. Задати точність розрахунків поточного елемента ряду;
  - 2.4. Задати початкову кількість елементів ряду, що дорівнює 1.
  - 2.5. Поки значення поточного елемента ряду більше за значення точності розрахунків повторювати такі дії:
    - визначити значення елемента ряду;
    - додати значення елемента ряду до суми;
    - перейти до наступного елемента ряду.
  - 2.6. Визначити похибку.
  - 2.7. Змінити поточне значення аргументу функції.
  - 2.8. Перейти до дій п. 2.2.
3. Кінець повторень

### Код:

```
#include <iostream>
#include <math.h>
using namespace std;
```

```
float accuracy;
```

```
//===== Функція розвитку cos в ряд Тейлора =====//
double cos_Taylor(double a) {
    double n = 1.0;    //номер елемента ряду
    double sum = 0.0;
    int i = 1;
    do {
        sum += n;
```

```

    n *= -1.0 * a * a / ((2 * i - 1) * (2 * i));
    i++;
} while (fabs(n) > accuracy);
return sum;
}

int main() {
    setlocale(LC_ALL, "Rus");

    double x, x1, xn, step, accuracy, y_Taylor = 0;

    cout << "Введите начальное значение функции : ";
    cin >> x1;
    if ((x1 - int(x1)) != 0) {
        cout << "\n/===== Error
=====^n";
        cout << " Начальное значение НЕ удовлетворяет условие\n";
        exit(0);
    }

    cout << "Введите конечное значение функции : ";
    cin >> xn;
    if (xn < x1 || (xn - int(xn)) != 0) {
        cout << "\n/===== Error
=====^n";
        cout << " Конечное значение НЕ удовлетворяет условие\n";
        exit(0);
    }

    cout << "Введите шаг функции : ";
    cin >> step;
    if (step > (xn - x1)) {
        cout << "\n/===== Error
=====^n";
        cout << " Шаг НЕ удовлетворяет условие\n";
        exit(0);
    }

    cout << "Введите точность расчётов для ряда Тейлора : ";
    cin >> accuracy;

    for (x = x1; x <= xn; x += step) {
        y_Taylor = cos_Taylor(x);
        cout << "-----\n";
        cout << "При X = " << x << "\nТейлор = " << y_Taylor << "\n";
    }
}

```



Результат:

```
Введите начальное значение функции : -5
Введите конечное значение функции : 5
Введите шаг функции : 1
Введите точность расчётов для ряда Тейлора : 0.001
```

```
-----
При X = -5
Тейлор = 0.283662
```

```
-----
При X = -4
Тейлор = -0.653644
```

```
-----
При X = -3
Тейлор = -0.989992
```

```
-----
При X = -2
Тейлор = -0.416147
```

```
-----
При X = -1
Тейлор = 0.540302
```

```
-----
При X = 0
Тейлор = 1
```

```
-----
При X = 1
Тейлор = 0.540302
```

```
-----
При X = 2
Тейлор = -0.416147
```

```
-----
При X = 3
Тейлор = -0.989992
```

```
-----
При X = 4
Тейлор = -0.653644
```

```
-----
При X = 5
Тейлор = 0.283662
```

Рис. 2. Виведений результат виконання завдання 2

## **Висновки:**

У ході виконання лабораторної роботи я дізналася про покажчики, посилання та створення нових функцій у мові програмування C++, їх особливості роботи та принцип застосування. Я навчилась використовувати функції, покажчики та посилання для розв'язування за допомогою них поставлених математичних задач.