Лабораторная работа 4.
Процедуры и функции.

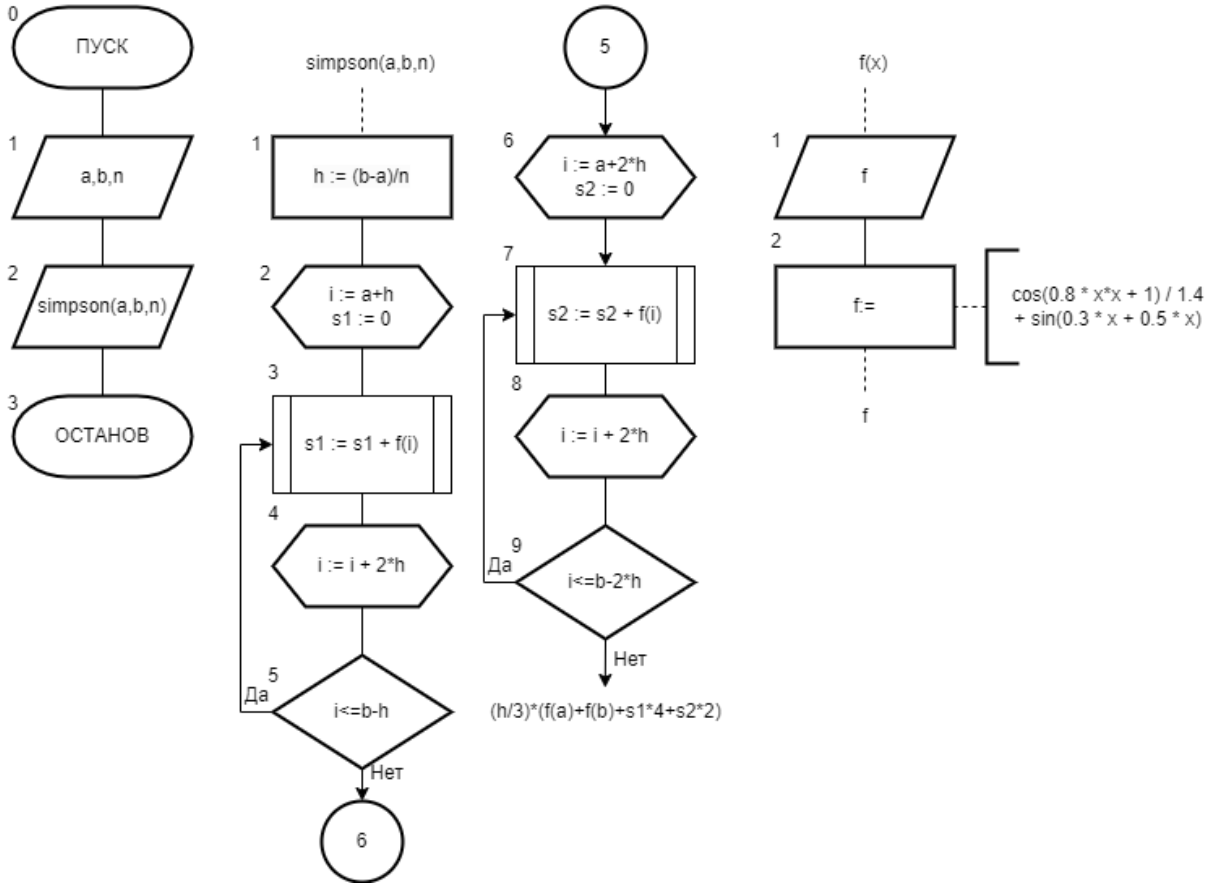
Цель работы: научится использовать процедуры и функции в среде разработки языка С.
Используемое оборудование: ПК, Visual Studio Community.

Задача 1: реализовать алгоритм на вычисление определенного интеграла методом парабол, используя пользовательскую функцию.

Математическая модель:

$$\int_a^b f(x)dx = \frac{h}{3}(f(a) + f(b) + 4 * \sum_{a+h}^{b-h} f(x) + 2 * \sum_{a+2h}^{b-2h} f(x))$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
a	double	Нижний предел интегрирования
b	double	Нижний предел интегрирования
n	double	Кол-во разбиений
f	double	Функция для вычисления функции
simpson	double	Функция для вычисления интеграла методом Симпсона
h	double	Шаг

s1	double	Промежуточные вычисления
s2	double	Промежуточные вычисления
i	double	Параметр цикла

Код программы:

```c
#include <stdio.h>
#include <math.h>
double f(double x)
{
    return cos(0.8 * x*x + 1) / 1.4 + sin(0.3 * x + 0.5 * x);
}
double simpson(double a, double b, double n)
{
    double h,s1,s2;
    s1 = 0;
    s2 = 0;
    h = (b - a) / n;
    for (double i = a+h; i <= b-h; i = i + 2*h)
    {
        s1 += f(i);
    }
    for (double i = a+2*h; i < b-2*h; i = i + 2*h)
    {
        s2 += f(i);
    }
    return (h / 3) * (f(a) + f(b) + s1*4 + s2*2);

}

int main()
{
    double a, b, n;
    printf("Enter the lower integration limit (a):\n");
    scanf_s("%lf", &a);
    printf("Enter the upper integration limit (b):\n");
    scanf_s("%lf", &b);
    printf("Enter the number of splits (n):\n");
    scanf_s("%lf", &n);
    printf("%lf", simpson(a,b,n));
}
```

Результат работы программы:

```
Enter the lower integration limit (a):
0.4
Enter the upper integration limit (b):
1.4
Enter the number of splits (n):
10
0.522998
```

```
Enter the lower integration limit (a):
0.4
Enter the upper integration limit (b):
1.4
Enter the number of splits (n):
10000
0.551460
```

```
Enter the lower integration limit (a):
0.4
Enter the upper integration limit (b):
1.4
Enter the number of splits (n):
1000
0.550858
```

Анализ вычислений:
Введение отдельных функций сильно упрощает код, позволяя не менять его при повторении действий или для вычисления одного и того же, только с разными данными.
Вывод:
Я научился реализовывать вычисление интеграла используя функции и процедуры на языке С.

Задание 2. Вычислить **z = x1 + x2 + x3**.

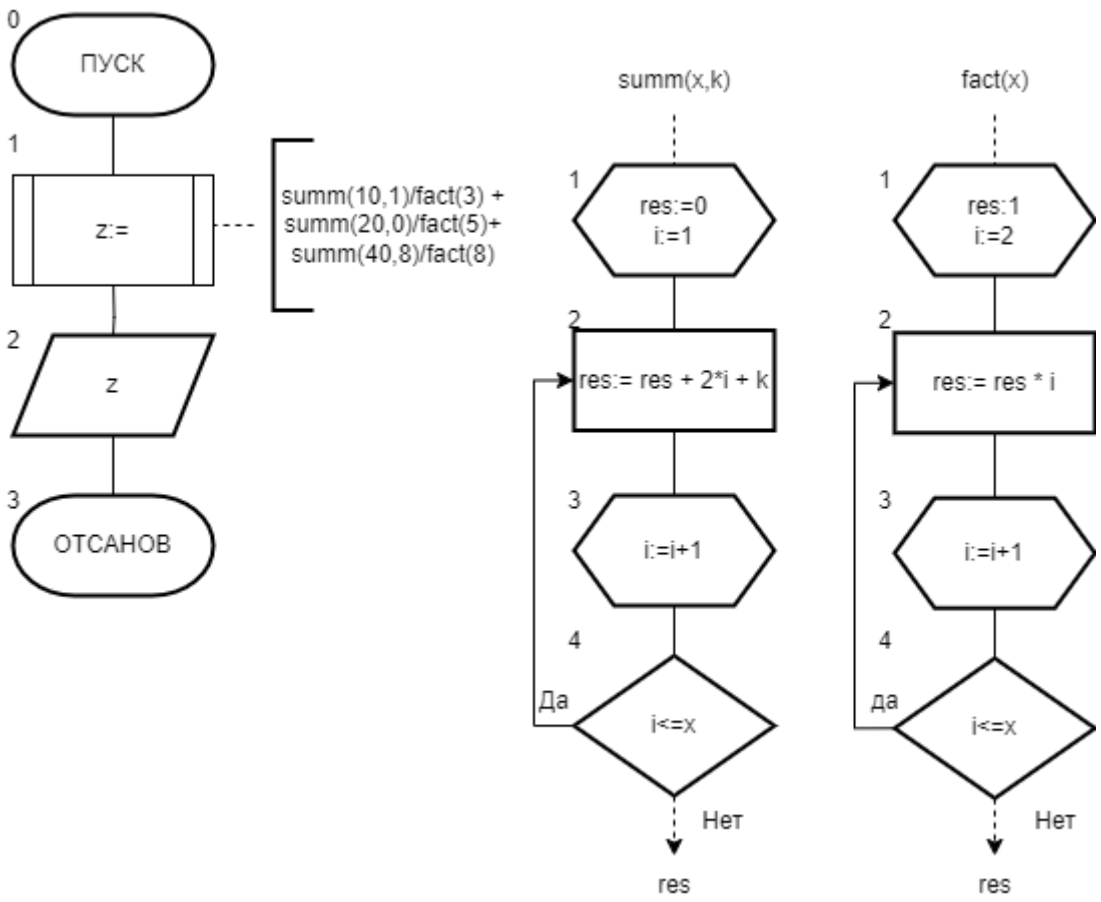
Математическая модель:

$$x_1 = \frac{\sum_{i=1}^{10}(2i+1)}{3!}$$

$$x_2 = \frac{\sum_{i=1}^{20} 2i}{5!}$$

$$x_3 = \frac{\sum_{i=1}^{40}(2i+3)}{8!}$$

Блок схема:



Список идентификаторов

z	double	Общий результат
summ	double	Функция, считающая сумму(сигма)
fact	double	Функция, считающая факториал
res	double	Промежуточный результат
i	double	Параметр цикла
x	double	Параметр функции

Код программы:

```
#include <iostream>
double fact(double x)
{
    double res = 1;
    for (double i = 2;i <= x; i++)
    {
        res *= i;
    }
    return res;
}
double summ(double x, double k)
{
    double res = 0;
    for (double i = 1; i <= x; i++)
    {
        res += 2 * i + k;
    }
    return res;
}
int main()
{
    double z = summ(10,1)/fact(3) + summ(20,0)/fact(5)+summ(40,8)/fact(8);
    printf("%lf", z);
}
```

Результат работы программы:

```
23.548611
```

Анализ вычислений:

Введение отдельных функций сильно упрощает код, позволяя не менять его при повторении действий или для вычисления одного и того же, только с разными данными.

Вывод:

Я научился реализовывать вычисление z используя функции и процедуры на языке C.