Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Марийский государственный университет»

Кафедра прикладной математики и информатики

Отчет по лабораторной работе № 1

на тему: **«Язык программирования C/С++. Структура программы. Система типов.»**

Выполнил: студент группы СИ-15 Елисеев Д.А.

Проверил: старший преподаватель Виноградов В.О.

Йошкар-Ола, 2023

Содержание

[Введение 3](#_Toc152830055)

[1. Задания лабораторной работы 4](#_Toc152830056)

[2.1.1 Разработка программы в виде блок-схемы 5](#_Toc152830057)

[2.1 Описание блок-схемы алгоритма 7](#_Toc152830058)

[3. Разработка программы в PascalABC 8](#_Toc152830059)

[3.1 Описание программного кода 10](#_Toc152830060)

[Заключение 12](#_Toc152830061)

# Введение

Лабораторная работа направлена на изучение возможностей C++ и систему типов. Приобретенные навыки в решении элементарных задач в C++ помогут в изучении курса по программированию.

# 1. Задания лабораторной работы

Разработать алгоритм, реализовать его в виде блок-схемы и запрограммировать–написать программный код, отладить его, запустить на выполнение, проверить на верность выводимые результаты.

1. Написать программу вывода на экран трех последовательно идущих чисел, каждое на отдельной строке. Первое число вводит пользователь, остальные числа вычисляются в программе.
2. Написать программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке.
3. Написать программу, вычисляющую объём куба и площадь его полной поверхности по введённому значению длины ребра.
4. Написать программу, в которой вычисляется сумма, разность и произведение двух целых чисел, введенных с клавиатуры.
5. Напишите программу для нахождения цифр четырёхзначного числа.
6. Напишите программу, вычисляющую значение заданной в варианте функции. Вариант выбирается по порядку следования в списке учащихся. (Вариант 2)

# 2. Разработка программы в виде блок-схемы

Блок-схема построена в редакторе RBS и изображена ниже на рисунках 1 и 2.

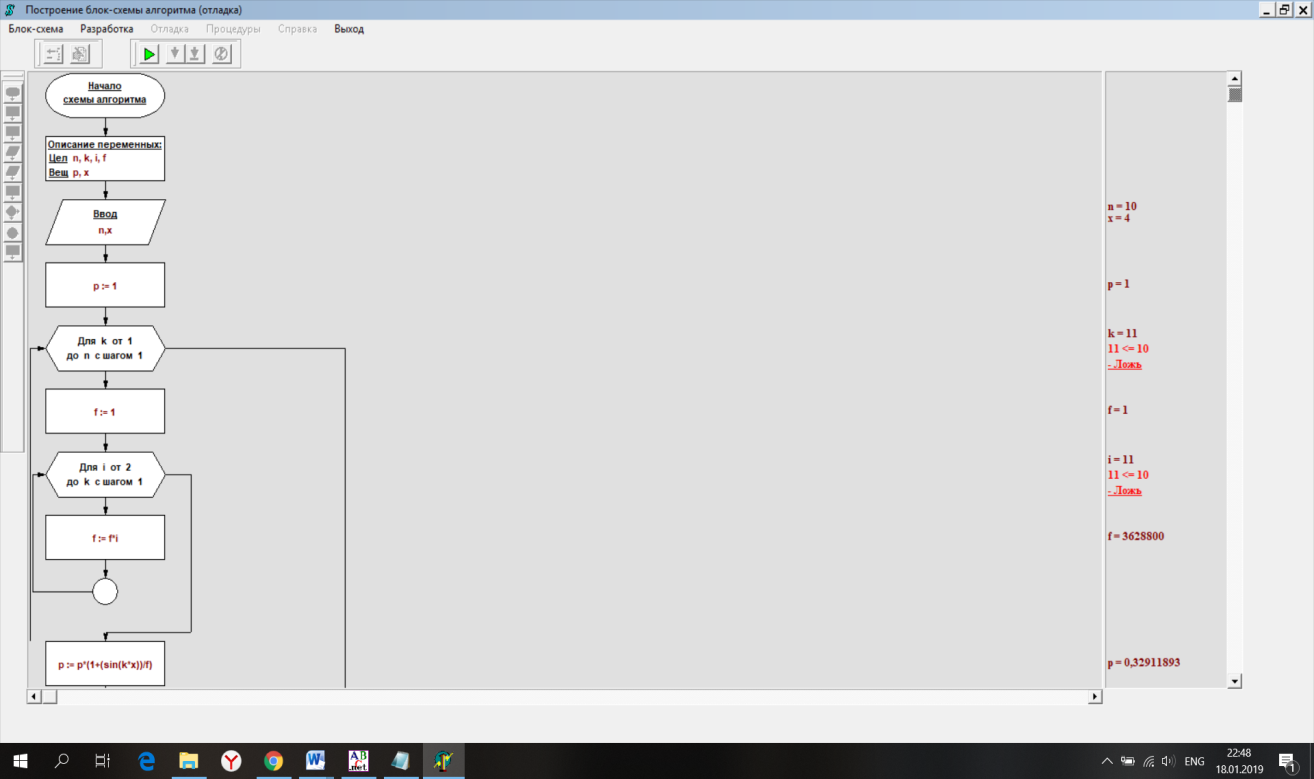


Рисунок 1 – Блок-схема алгоритма

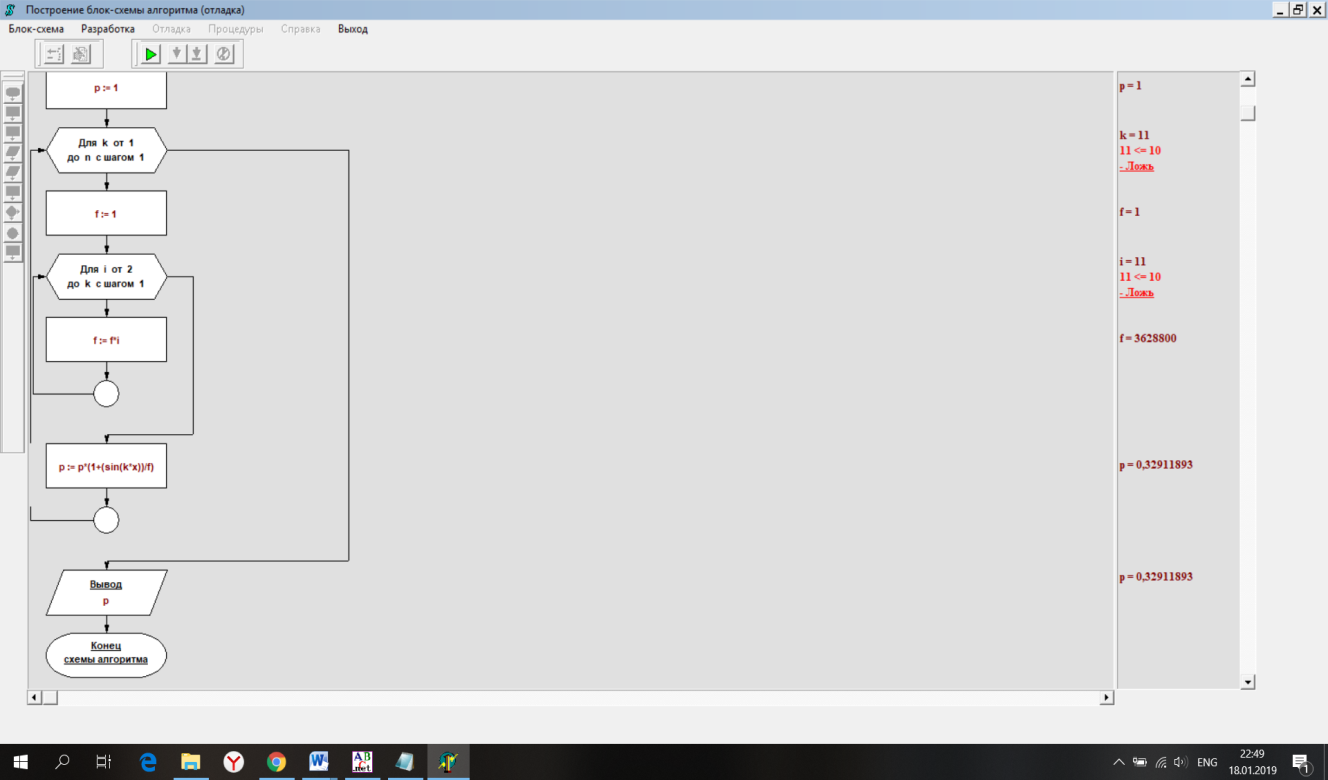


Рисунок 2 – Блок-схема алгоритма

## 2.1 Описание блок-схемы алгоритма

Сначала описываем переменные и их типы. В нашем случае переменные n, k, I, f – целого типа, а p, x – вещественного. Вводим значения переменных n, x, т.е. значение n – число, до которого будет вычисляться произведение, и x – переменная функции sin(x).

Затем присваиваем переменной p значение 1, для того, чтобы результат произведения не изменился, т.к. при умножении на 1 результат не меняется (2):

(2)

Далее используем счетчик «Для» k от 1 до n с шагом 1. Тогда, конечное значение n увеличится на 1. Он нужен для определения пределов вычисления значения данного выражения. Переменной f присваиваем значение 1, т.к. вычисление факториала начинается с нее (3):

(3)

Используем еще один цикл для i от 2 до k с шагом 1. Затем вычисляем факториал. Как только условие примет значение false, находим значение выражения (4):

(4)

Далее выводим значение p на экран и проверяем результат.

# 3. Разработка программы в PascalABC

**var**

n:integer;

x:real;

**procedure** vvod(**var** n:integer; **var** x:real);

**begin**

write('n=');

read(n);

write('x=');

read(x);

**end**;

**function** fx(x:real; n:integer):real;

**var** p,f:real;

**begin**

p:=1;

**for var** k:=1 **to** n **do begin**

f:=1;

**for var** i:=2 **to** k **do**

f:=f\*i;

p:=p\*(1+(sin(k\*x))/f);

**end**;

fx :=p;

**end**;

**procedure** vivod;

**begin**

writeln('p=',fx(x,n));

**end**;

**begin**

vvod(n,x);

vivod;

**end**.

Запуск на выполнение и проверка выводимых результатов представлена ниже на рисунке 3.

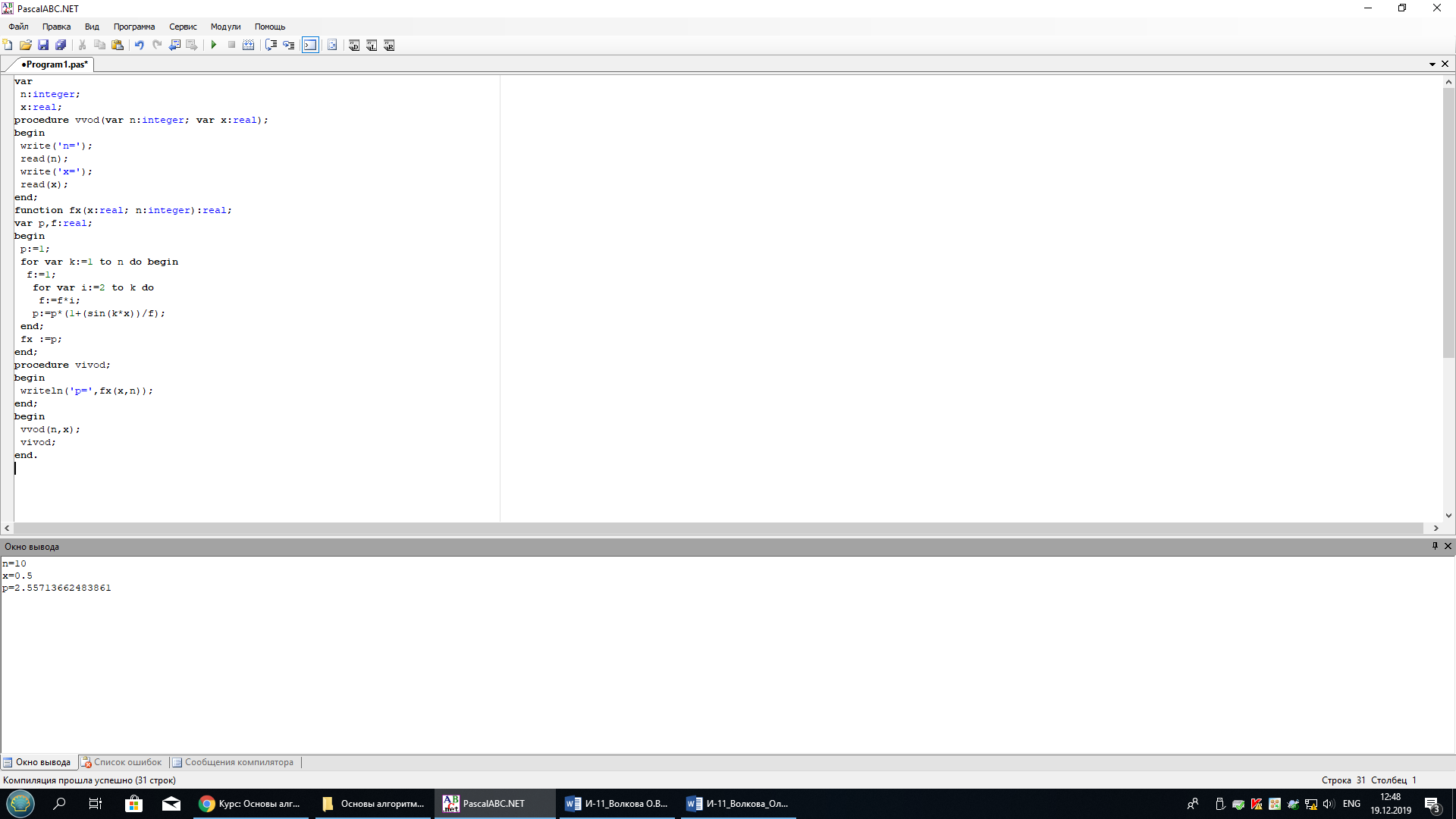


Рисунок 3 – Запуск программы на выполнение

## 3.1 Описание программного кода

Сначала объявляем одну переменную вещественного типа и одну переменную целого типа. В нашем случае переменная n – это переменная целого типа, а x – вещественного:

**var**

n:integer;

x:real;

Далее вводим подпрограмму, а точнее процедуру ввода, и просим пользователя ввести значения переменных n и x:

**procedure** vvod(**var** n:integer; **var** x:real);

**begin**

write('n=');

read(n);

write('x=');

read(x);

**end**;

Затем вводим подпрограмму функции fx. Присваиваем переменной p начальное значение, равное 1, чтобы ответ не менял свое значение. Устанавливаем цикл «Для» k:=1 от 1 до n с шагом 1 и цикл начинает свою работу. Там мы присваиваем переменной f значение 1, т.к. факториал – это произведение последовательных натуральных чисел, начиная с единицы. Тут же вводим еще один цикл «Для» i:=2 от 2 до k с шагом 1. Здесь же мы находим значение факториала, и находим значение нашей функции:

**function** fx(x:real; n:integer):real;

**var** p,f:real;

**begin**

p:=1;

**for var** k:=1 **to** n **do begin**

f:=1;

**for var** i:=2 **to** k **do**

f:=f\*i;

p:=p\*(1+(sin(k\*x))/f);

**end**;

fx :=p;

**end**;

Вводим еще одну процедуру, только теперь уже процедуру вывода, где выводим значение функции:

**procedure** vivod;

**begin**

writeln('p=',fx(x,n));

**end**;

На последнем этапе вызываем процедуру ввода и процедуру вывода:

**begin**

vvod(n,x);

vivod;

**end**.

Конец программы.

# Заключение

Данную задачу считаю решённой, так как все условия задачи выполнены. А именно: разработан алгоритм, разработанный алгоритм представлен в виде блок-схемы, задача запрограммирована, написан программный код в среде PascalABC.NET. Кроме того, программа запущена на выполнение.