



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«МИРЭА – Российский технологический университет»
РТУ МИРЭА**

Институт искусственного интеллекта

Кафедра системной инженерии

Отчёт по практической работе № 6

**по дисциплине «Программное и алгоритмическое обеспечение систем
сбора и обработки данных»**

Выполнил студент группы КСБО-02-23

Грязцов В.А.

Проверил доцент кафедры системная инженерия

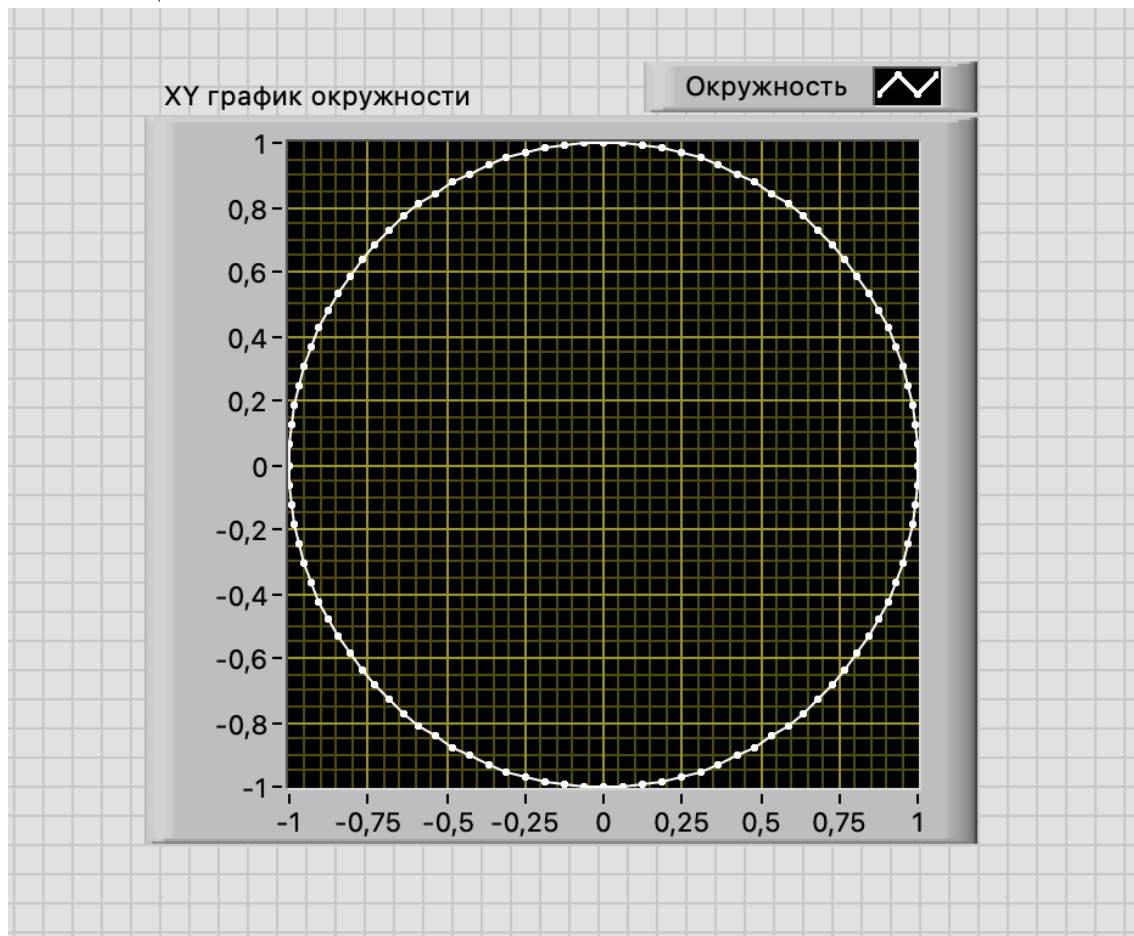
Мошкин В.В.

Москва 2024

Задание 1. ВП График окружности

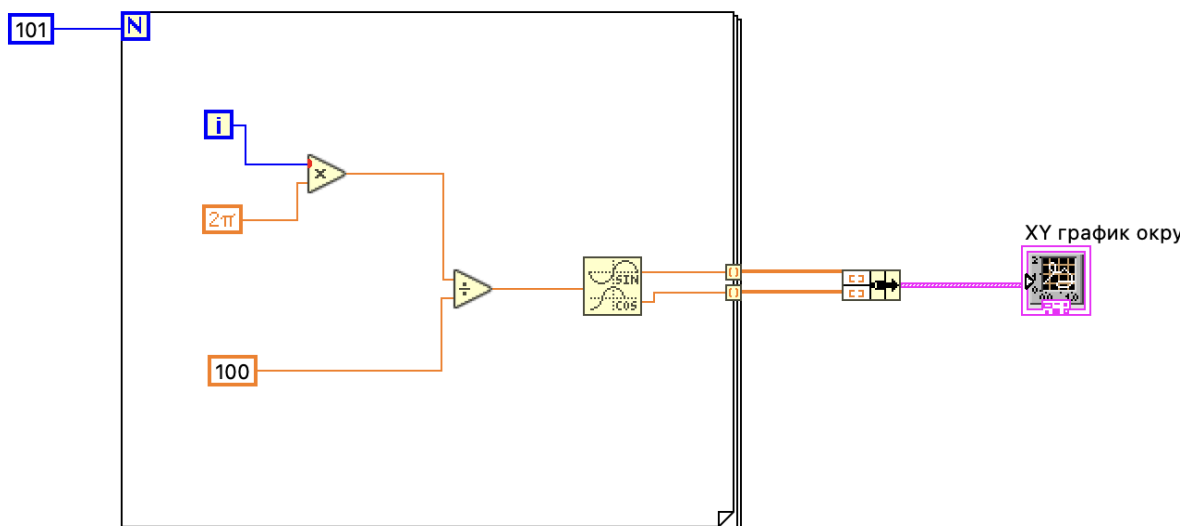
Цель: Построить осциллограмму окружности, используя двухкоординатный график Осциллограмм.

Лицевая панель



На лицевой панели расположен **XY Graph**, на котором при запуске программы отрисовывается окружность с радиусом 1. Точки графика выполнены в форме маленьких квадратов.

Блок-диаграмма



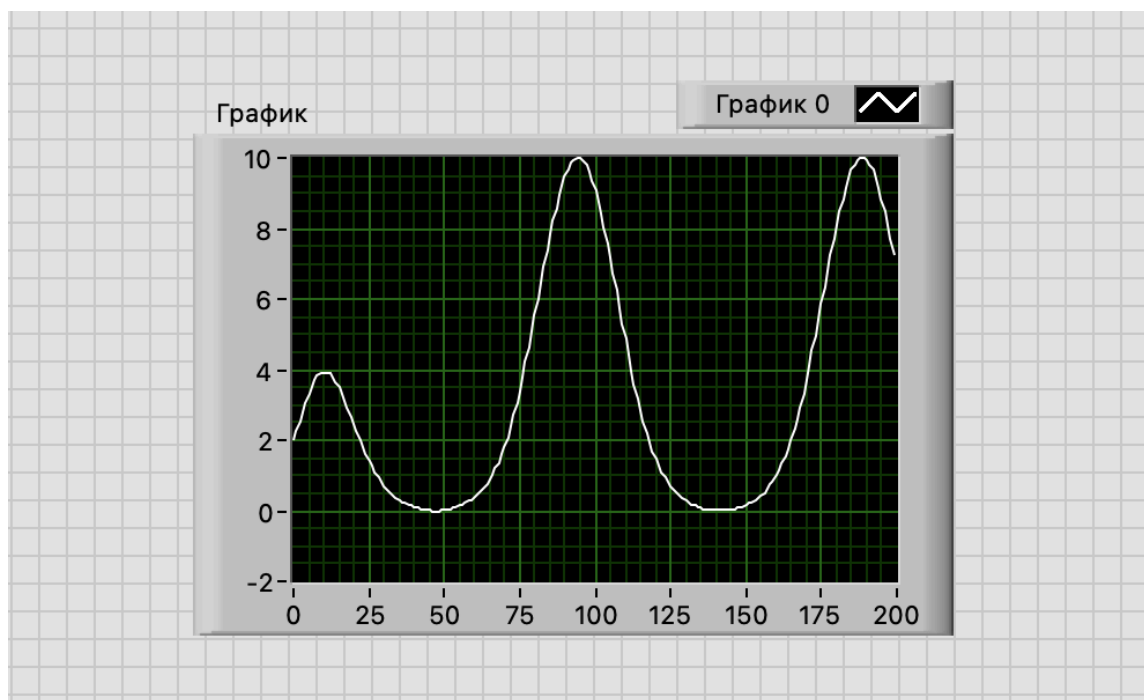
В цикле **For Loop**, который выполняется 101 раз, номер итерации умножается на 2π и делится на 100, полученное число передается на функцию **Sine&Cosine** (функция вычисления синусов и косинусов), затем массивы синуса и косинуса объединяются при помощи **Bundle** и выводится на графике.

Вывод: функция вычисляет значения синусов и косинусов и объединяет эти значения, в результате вывода которых получается график окружности.

Задание 2. ВП Узел Формулы

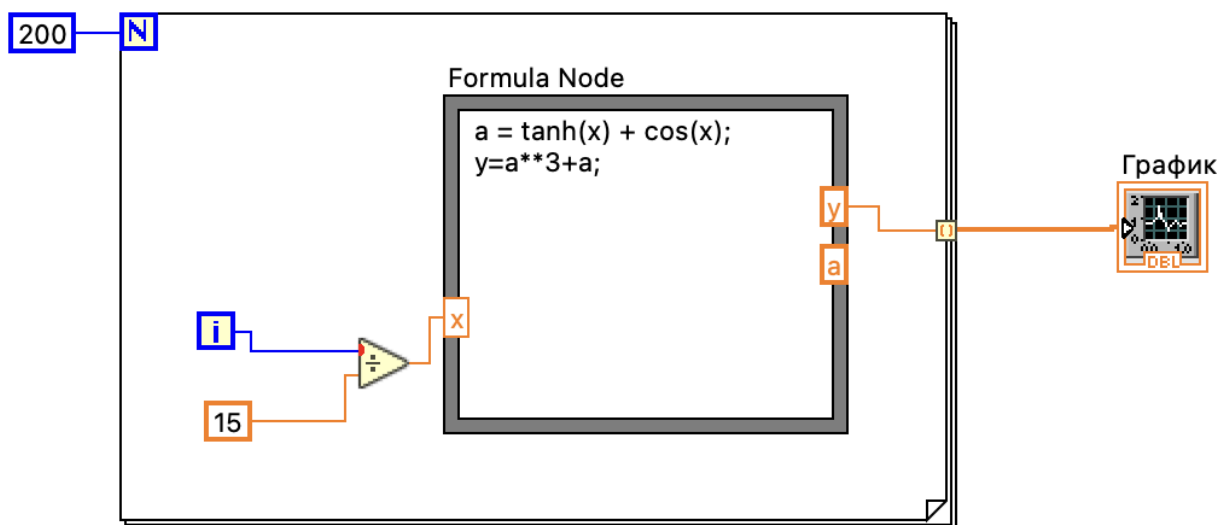
Цель: Изучить работу структуры узел Формулы.

Лицевая панель



На лицевой панели расположен **Waveform Graph**, шкала **X** имеет значения от 0 до 200, шкала **Y** от -2 до 10.

Блок-диаграмма



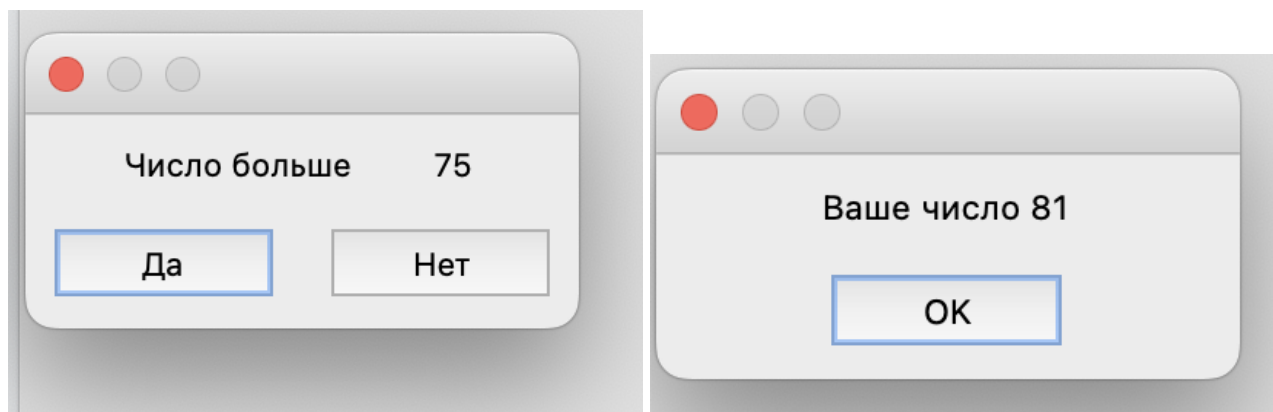
На блок-диаграмме в цикле **For Loop**, который выполняется 200 раз, значение номера итерации делится на 15 и передается в **Formula Node** (как переменная x), где вычисляется $a = \tanh(x) + \cos(x)$ и $y = a^{**}3 + a$, полученный y выводится на графике.

Вывод: **Formula Node** позволяет довольно просто производить вычисления функций, которые затем используются в дальнейшей работе виртуального прибора.

Задание 3. ВП Угадай число

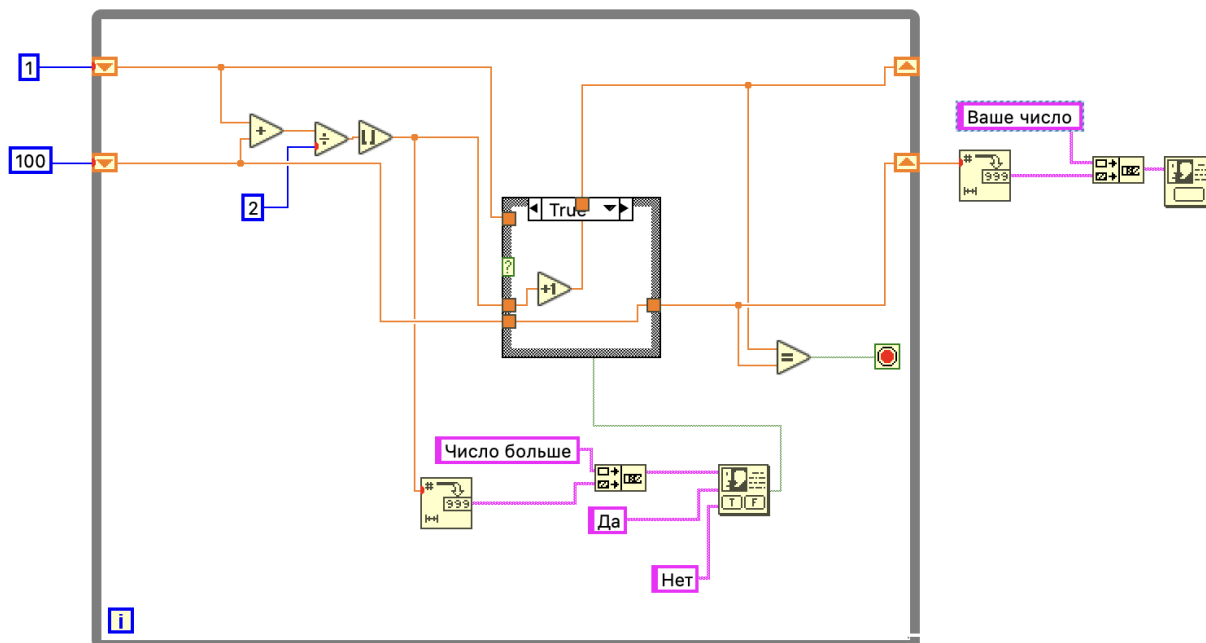
Цель: Создать ВП, который угадывает загаданное число (от 1 до 100) максимум за семь шагов.

Лицевая панель



Взаимодействие с виртуальным прибором проходит только через всплывающее окно, на котором, отображается вопрос и угаданное число.

Блок-диаграмма



На блок-диаграмме в цикл **For Loop** передаются нижний и верхний пределы числа, которые изменяют свои значения при помощи **Shift Register**, затем считается среднее значение между ними и округляется в меньшую сторону, полученное среднее значение переводится в строку и конкатенируется со строкой “Число больше”, объединенная строка выводится на **Two Button Dialog**, ответ Да соответствует значению **True**, а Нет - **False**. Если получен **True**, то в **Case Structure** среднее значение увеличивается на 1 и становится нижней

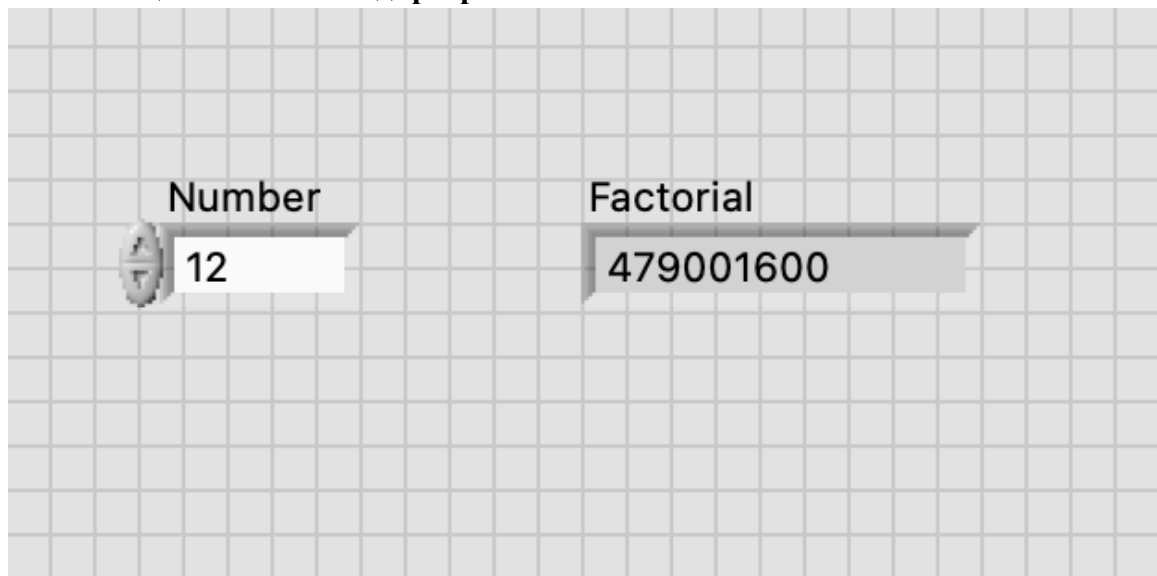
границей, при **False** верхняя граница получает среднее значение, когда верхняя и нижняя границы будут равны цикл завершится и будет выведен ответ.

Вывод: Алгоритм выполняет бинарный поиск, сужая диапазон чисел до тех пор, пока нижняя и верхняя границы не совпадут. Среднее значение между ними проверяется с помощью диалога с пользователем, и в зависимости от ответа границы корректируются, пока не будет найдено искомое число.

Задание 4. ВП Число сочетаний

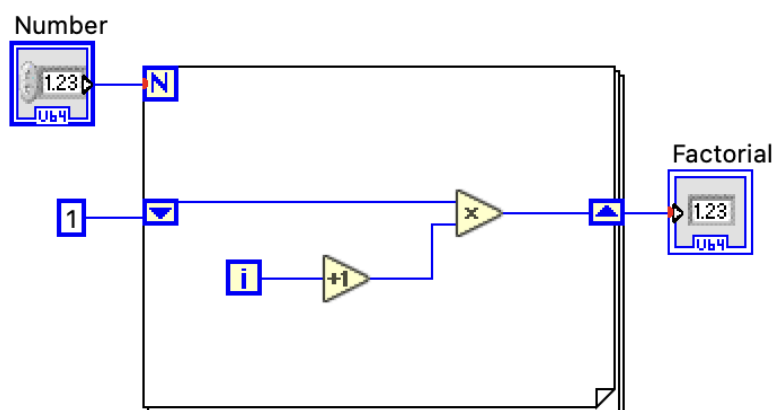
Цель: Создать ВП, вычисляющий факториал числа $k!$, и оформить его в качестве подпрограммы. Создать новый ВП, вычисляющий с помощью этой подпрограммы количество сочетаний для заданных параметров «m» и «n» и выводящий полученное значение на экран.

Лицевая панель подпрограммы



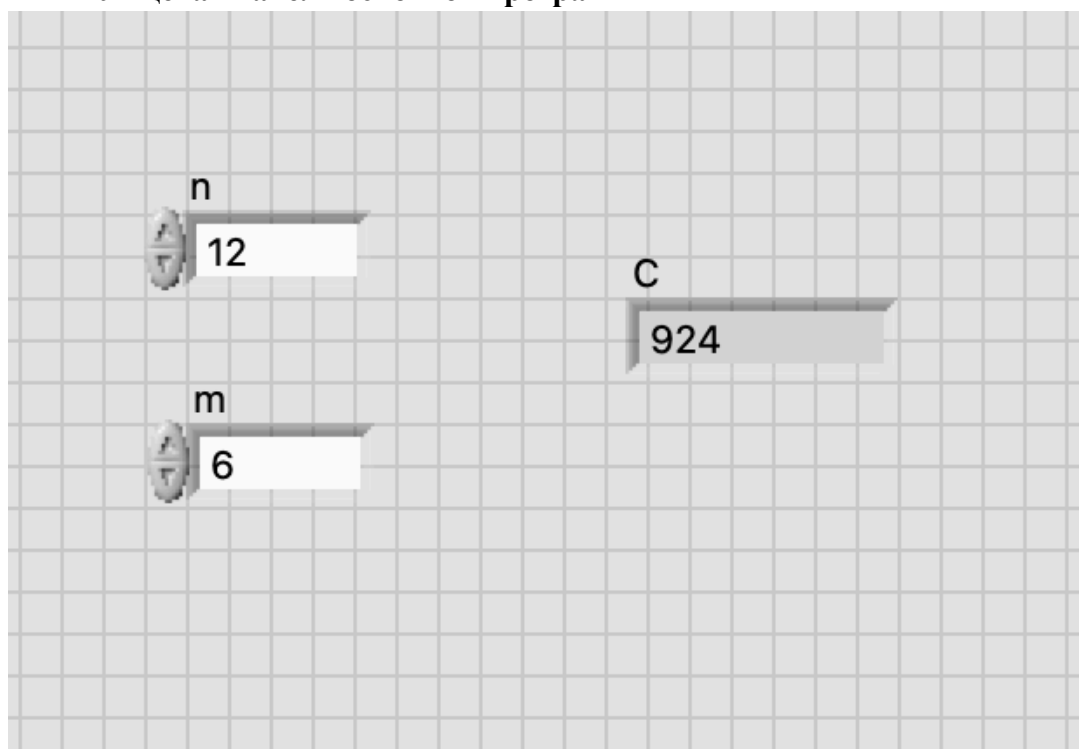
При помощи **Numeric Control** вводится число, для которого необходимо найти факториал, **Numeric Indicator** выводит этот факториал.

Блок-диаграмма подпрограммы



В цикле **For Loop** номер итерации увеличивается до **N** и умножается, на предыдущий результат выполнения цикла, который сохраняется в переменной при помощи **Shift Register**. Число, полученное в результате работы цикла передается на **Numeric Indicator**.

Лицевая панель основной программы

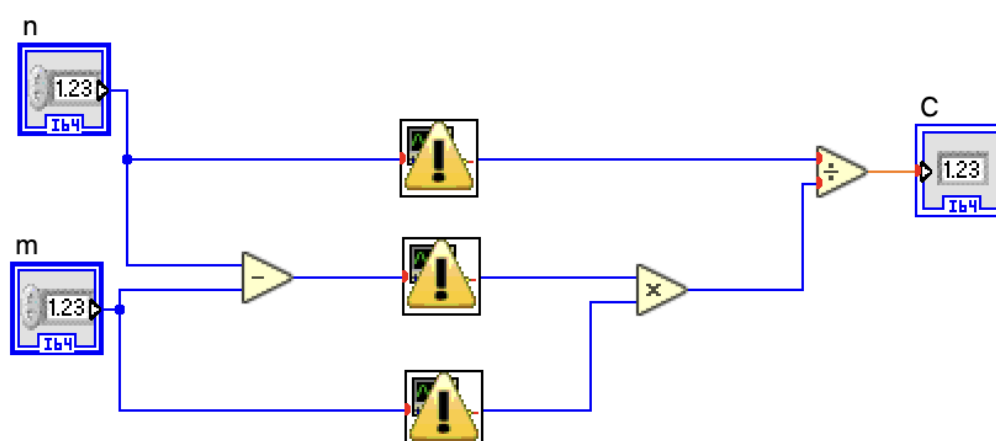


При помощи **Numeric Control** вводятся числа **m** и **n**, для которых вычисляется количество сочетаний по формуле:

$$C_n^m = \frac{n!}{(n-m)!m!}$$

Numeric Indicator выводит полученное количество сочетаний.

Блок-диаграмма программы



Значения **n**, **m** и **(n-m)** передаются в подпрограмму вычисления факториалов, факториал **(n-m)** умножается на **m**, затем **n** делится на **m** и выводится ответ.

Вывод: ВП позволяет производить вычисление количества сочетаний для заданных параметров **n** и **m** и выводит его на экран.