



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

Отчет

по лабораторной работе № 3

Название лабораторной работы: Программирование циклического процесса. Типы циклов.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-14Б

30.09.2023

А.С. Воеводин

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

30.09.2023

О.А. Веселовская

(Подпись, дата)

(И.О. Фамилия)

Цель работы – Рассмотреть решение предложенной задачи с использованием всех трёх видов циклов. Ознакомиться с каждым из них и выбрать наиболее рациональный, удобный для использования. Объяснить тот или иной выбор.

Задание – Написать программу, считающую сумму первых k чисел последовательности Фибоначчи.

Ход работы:

- Написание алгоритма для трёх разных видов цикла.
- Тестирование программы при различных k .
- Выбор лучшего из циклов и пояснение.
- Изображение программы с разными циклами в схеме алгоритма в Draw.io.
- Вывод.

Для начала напишем алгоритм с использованием счётного цикла for:

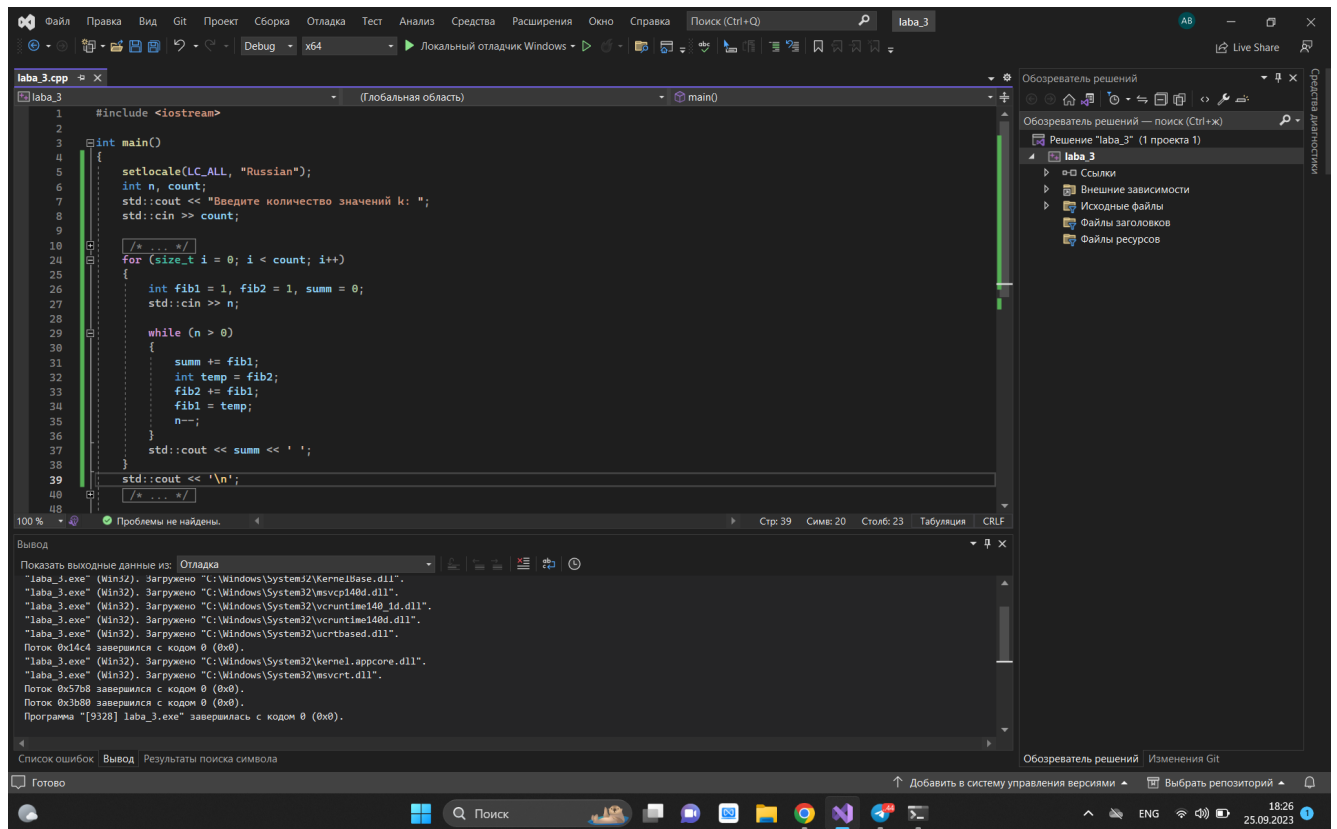


Рисунок 3 – Использование цикла-пока в программе

Проверим эту программу на тех же k:

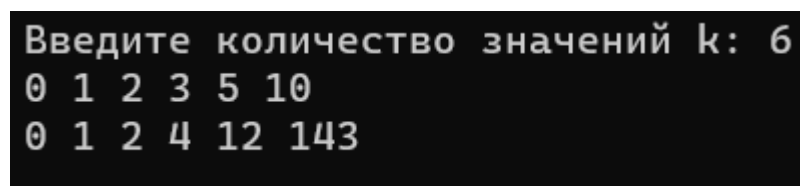


Рисунок 4 – Вывод суммы первых членов при разных k

Последним циклом идёт цикл-до, так будет выглядеть программа с его использованием:

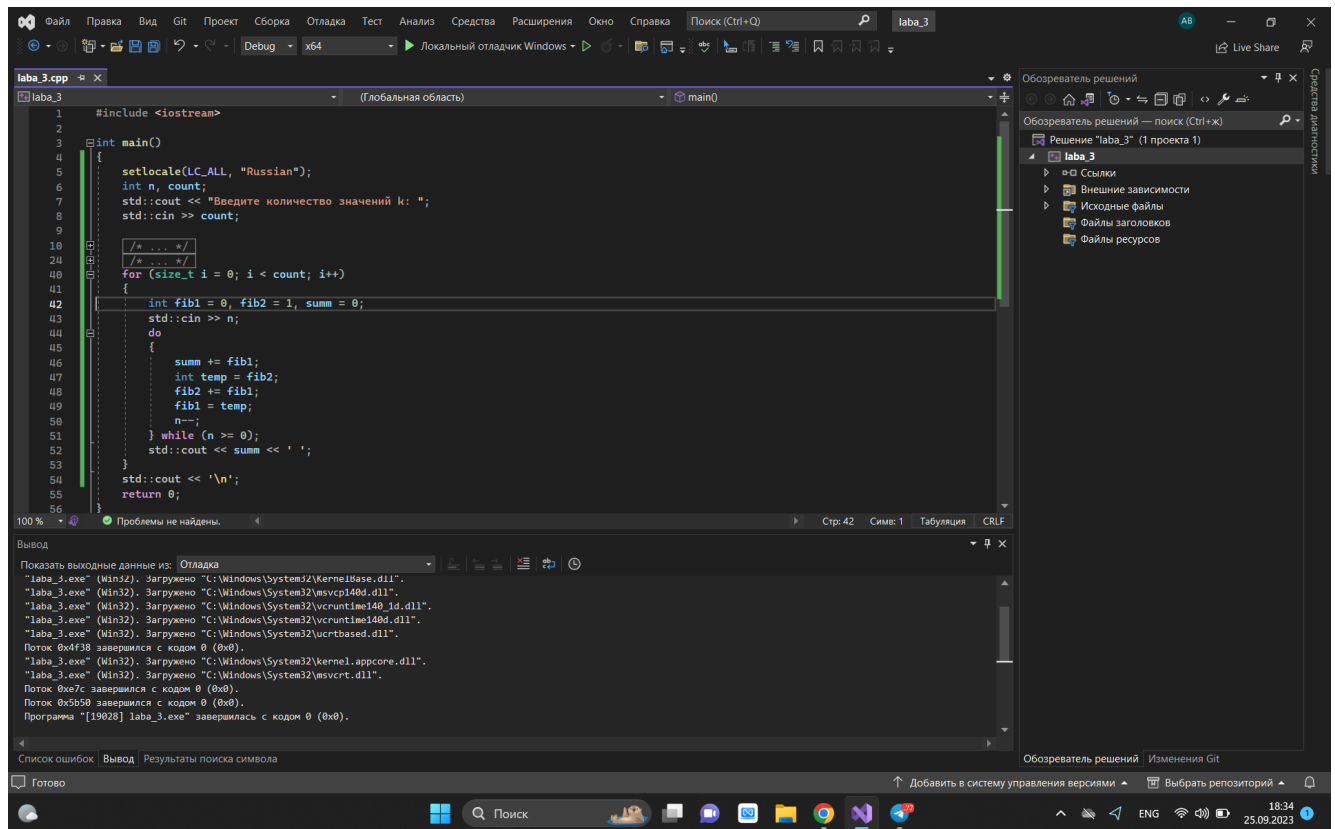


Рисунок 5 – Использование цикла-до в программе

Цикл-до отличается от цикла-пока и счётного тем, что при любом k сначала будет выполнено действие, и только потом проверка условия, поэтому чтобы при $k = 0$ программа вывела 0, можно либо добавить проверку для данного случая, либо, как в этой программе, изначально первый член задать 0, чтобы при сложении к сумме ничего не прибавлялось. За счёт этого действия, мы теряем одну итерацию, и чтобы её восполнить, в условии цикла пишем $n \geq 0$ либо $n > -1$. Проверим данную программу на тех же самых k :

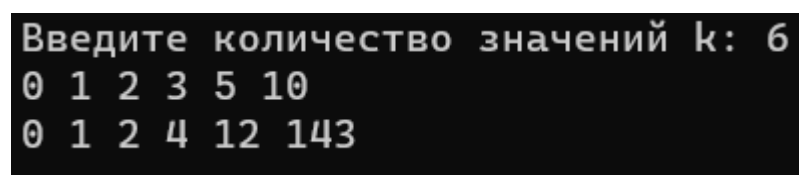


Рисунок 6 – Вывод суммы первых членов при разных k

Как видно из тестов, все три цикла справляются с поставленной задачей. Разница в первых двух циклах минимальна, по сравнению с третьим, так как из-за его особенностей приходится переписывать алгоритм, добавлять дополнительные проверки. Пользователю неважно, как написана программа, если результат правильный, но разработчику удобство написания программы и её отладка очень важны, поэтому считаю, что для данного алгоритма рациональным решением

будет использование счётного цикла, так как во-первых инициализация счётчика происходит внутри области видимости цикла, в отличие от других циклов, что немного экономит память и не засоряет дальнейшее написание программы ненужными переменными, а во-вторых условие остановки и шаг задаются при вызове этого цикла, что несомненно улучшает читабельность для разработчика. Изобразим эти программы в виде схемы алгоритма в Draw.io:

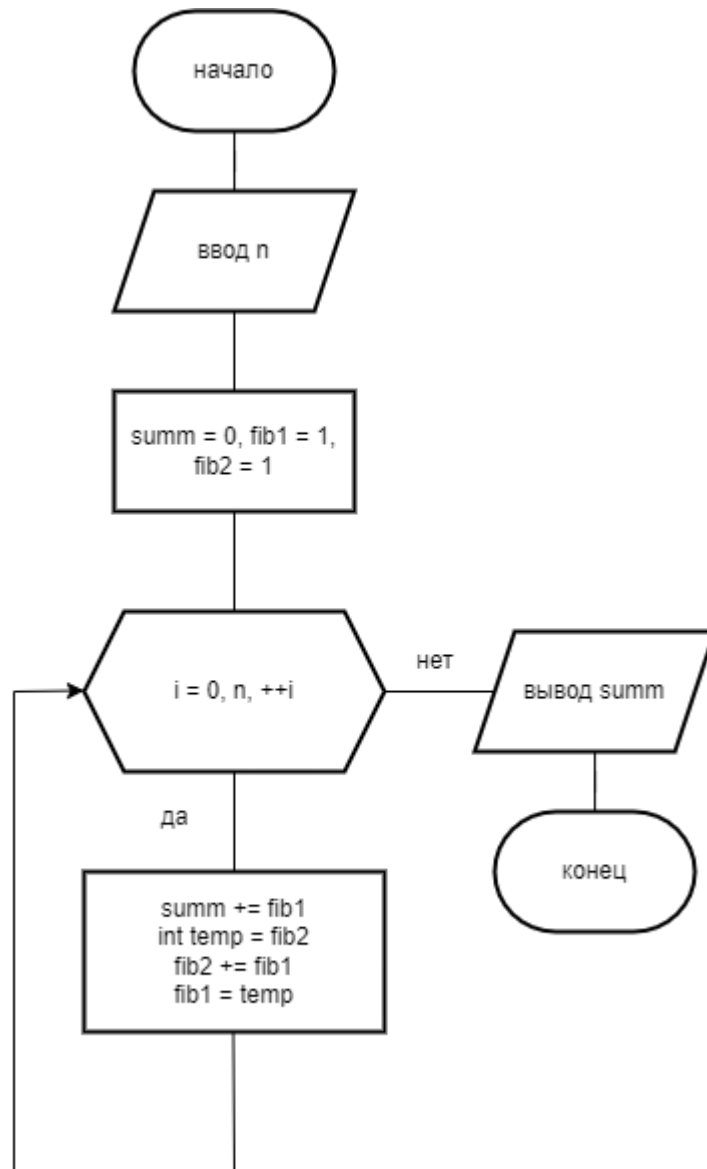


Рисунок 7 – Счётный цикл

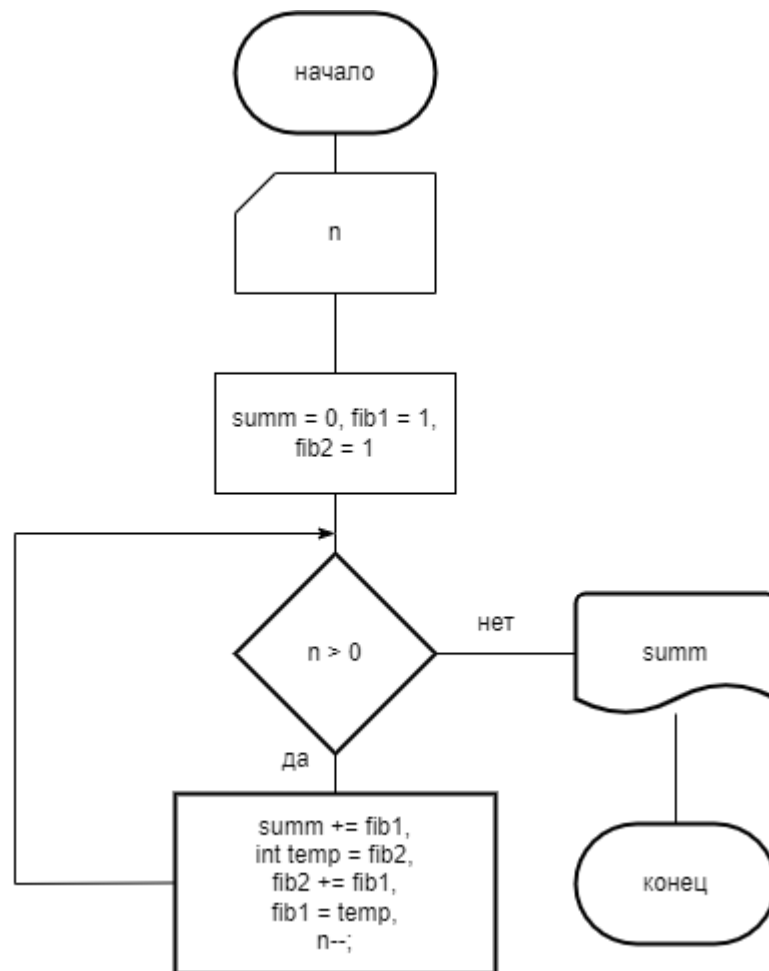


Рисунок 8 – Цикл-пока

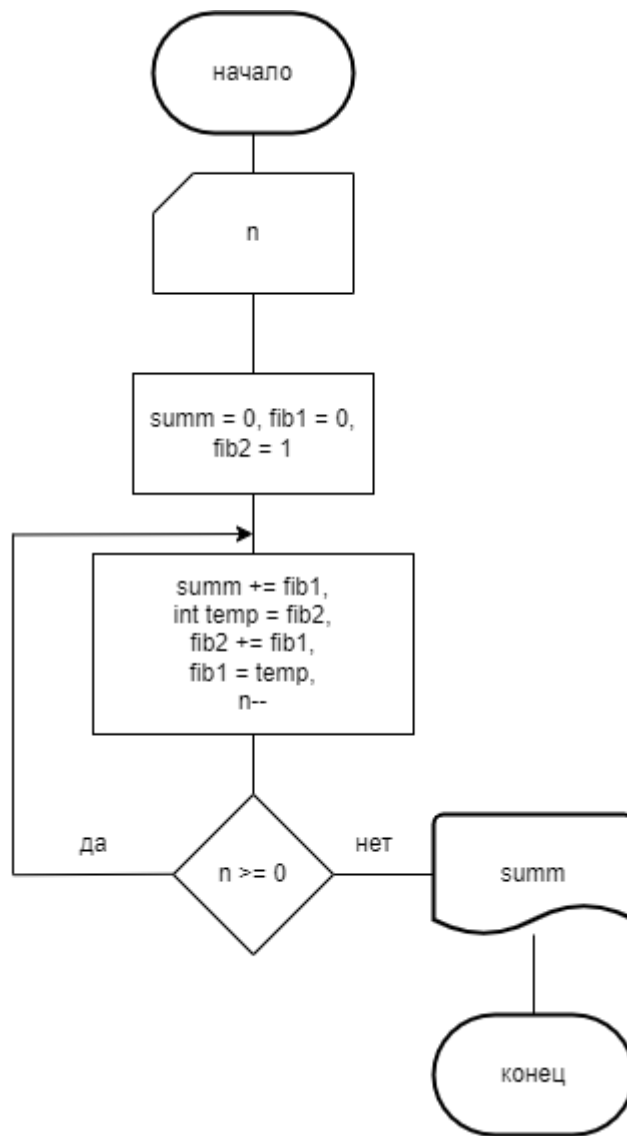


Рисунок 9 – Цикл-до

Вывод: В ходе лабораторной работы я научился работать с разными видами циклов, таких как счётный, цикл-пока и цикл-до, понял особенности реализации каждого из них, выбрал наиболее рациональный для решения поставленной задачи. Научился считать сумму первых k чисел Фибоначчи, отлаживать программу в циклических процессах, изображать циклические процессы в графических редакторах в виде схемы алгоритма.