**Отчет по лабораторной работе №3**

На тему «Анализ алгоритмов»

Вариант №17

Выполнил: ст.группы

2221121

Хайрутдинов А.А.

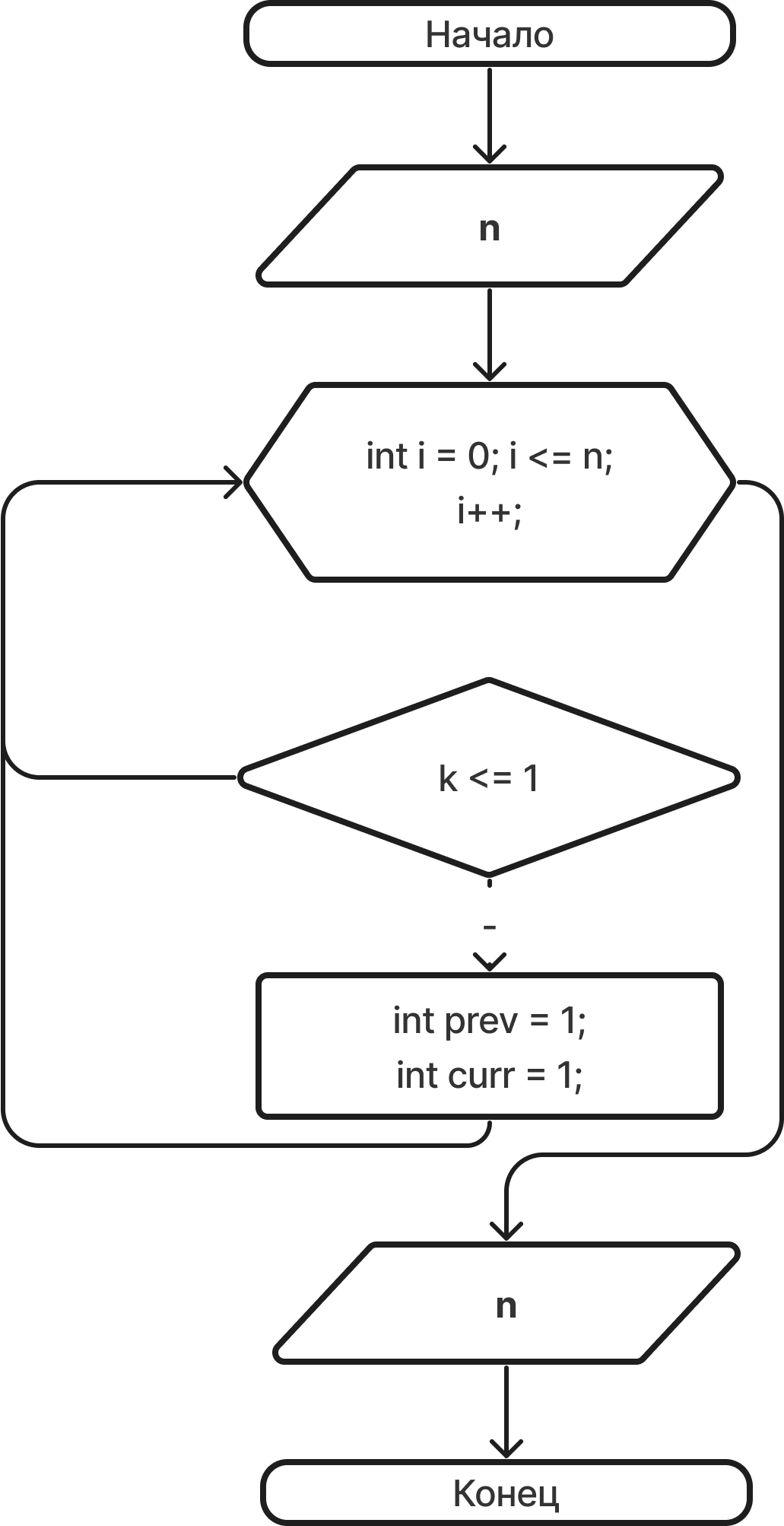
Проверил:

Хузятова Л.Б.

**Задание 1.1**

Предположим, на одной и той же машине проводится сравнительный анализ реализаций двух алгоритмов сортировки, работающих по методу вставок и по методу слияния. Для сортировки n элементов методом вставок необходимо 8n2 шагов, а для сортировки методом слияния – 64n lg n шагов. При каком значении n время сортировки методом вставок превысит время сортировки методом слияния?

**Блок-схема**

****

**Код программы**

class Program

{

// Метод для вычисления времени сортировки методом вставок для n элементов

static double InsertionSortTime(int n)

{

// Формула для времени сортировки методом вставок: 8 \* n^2

return 8 \* n \* n;

}

// Метод для вычисления времени сортировки методом слияния для n элементов

static double MergeSortTime(int n)

{

// Формула для времени сортировки методом слияния: 64 \* n \* log2(n)

return 64 \* n \* Math.Log(n, 2);

}

static void Main(string[] args)

{

// Начальное значение n. Количество элементов, которые требуется отсортировать

int n = 10;

// Увеличиваем значение n до тех пор, пока время сортировки методом вставок не превысит время сортировки методом слияния

while (InsertionSortTime(n) <= MergeSortTime(n))

{

n++;

}

// Выводим результат, когда время сортировки методом вставок превысит время сортировки методом слияния

Console.WriteLine($"Время сортировки методом вставок превысит время сортировки методом слияния при n = {n}");

}

}

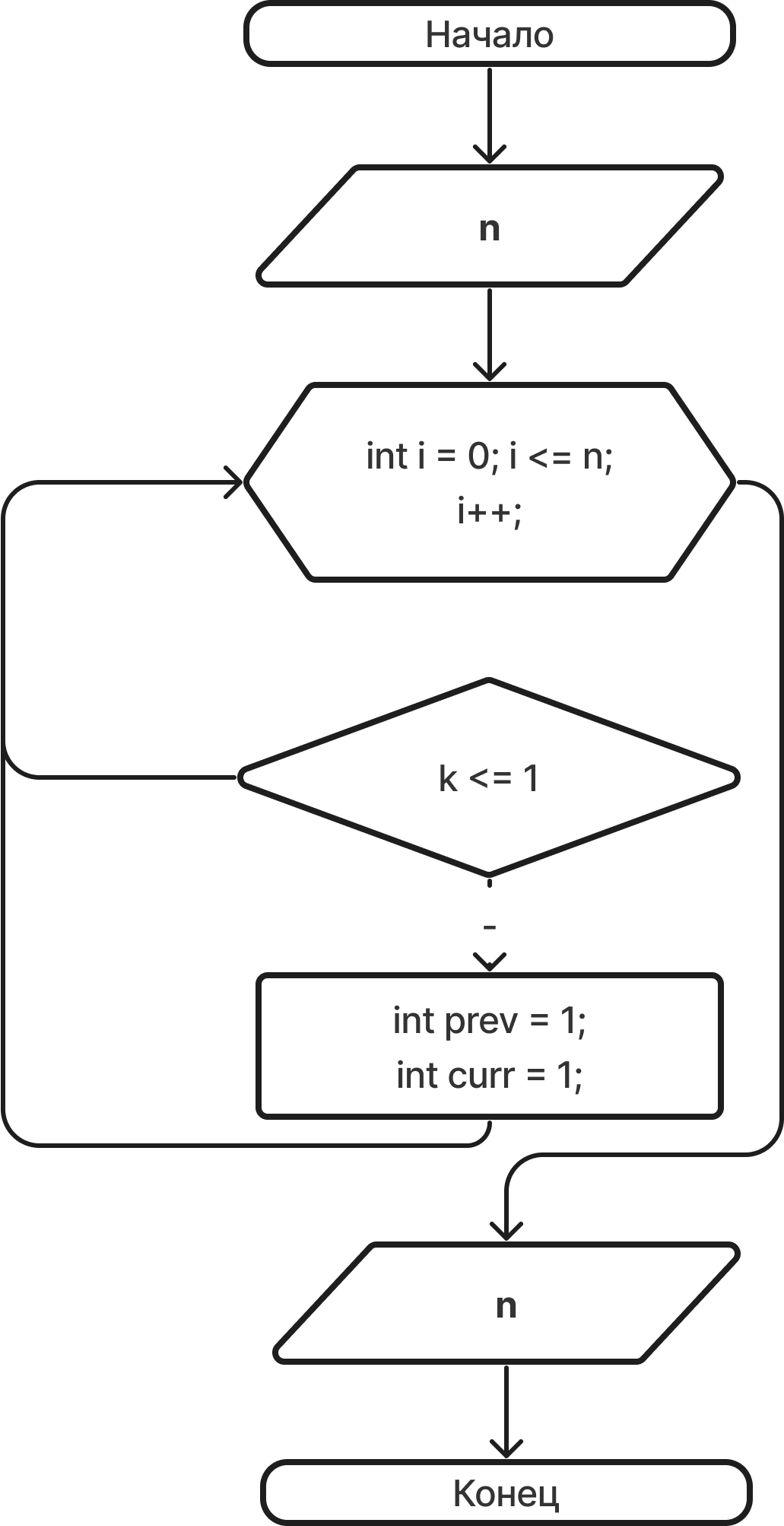
**Результат**



**Здание 5.8**

Имеется n бактерий красного цвета. Через 1 такт времени красная бактерия меняется на зеленую, затем через 1 такт времени делится на красную и зеленую. Определить, сколько будет всех бактерий через k тактов времени?

**Блок-схема**

****

**Код программы**

public class BacteriaSimulation

{

/// <summary>

/// Вычисляет сумму последних двух чисел в последовательности Фибоначчи до заданного номера.

/// </summary>

/// <param name="k">Номер последнего числа в последовательности Фибоначчи, для которого нужно вычислить сумму.</param>

/// <returns>Сумма последних двух чисел в последовательности Фибоначчи.</returns>

private static int CalculateFibonacci(int k)

{

// Если номер меньше или равен 1, возвращаем само число, так как это будет одно из двух последних чисел в последовательности.

if (k <= 1)

return k;

// Инициализация предыдущего и текущего чисел Фибоначчи.

int prev = 1;

int curr = 1;

// Проходим по последовательности, начиная с третьего числа (индекс 2) до заданного номера k.

for (int i = 2; i < k; i++)

{

// Вычисляем следующее число Фибоначчи путем сложения предыдущего и текущего чисел.

int next = prev + curr;

// Обновляем значения предыдущего и текущего чисел для следующей итерации.

prev = curr;

curr = next;

}

// Возвращаем сумму последних двух чисел Фибоначчи.

return prev + curr;

}

public static int CalculateTotalBacteria(int n, int k)

{

return CalculateFibonacci(k)\*n;

}

}

static void Main(string[] args)

{

int n = 2; // Изначальное количество бактерий

int k = 8; // Количество тактов времени

int totalBacteria = BacteriaSimulation.CalculateTotalBacteria(n, k);

Console.WriteLine($"Всего бактерий через {k} тактов времени: {totalBacteria} бактерий");

}

**Результат**

****

**Заключение**

В ходе лабораторной работы по анализу алгоритмов были рассмотрены основные понятия, связанные с трудоемкостью алгоритмов, рекуррентными уравнениями и асимптотическим анализом. Мы определили трудоемкость алгоритма как функцию от размерности задачи, которая оценивает сверху время, требуемое для ее решения, а также рассмотрели различные типы алгоритмов в зависимости от их временной сложности. лабораторная работа позволила ознакомиться с основными методами анализа алгоритмов и научиться оценивать их временную сложность, что является важным инструментом при выборе наиболее эффективного алгоритма для решения конкретной задачи.