

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины «Алгоритмизация»
Вариант _____

Выполнил:
Ивашенко Олег Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.02 «Информационные и
вычислительные машины»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики:
Доцент кафедры инфокоммуникации
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Тема: Сравнение наивного кода с продуманным

Порядок выполнения работы:

```
using System;

public class HelloWorld
{
    public static long Counter = 0;

    public static void Main(string[] args)
    {
        Console.WriteLine("[Program] Введите число: ");
        int N = int.Parse(Console.ReadLine());
        var Timer = DateTime.Now;
        Console.WriteLine("[Program] Результат: " + FibRecursive(N).ToString());
        Console.WriteLine("[Program] Количество выполнений: " + Counter.ToString());
        Console.WriteLine("[Program] Время выполнения: " + DateTime.Now.Subtract(Timer).TotalSeconds + " сек.");
    }

    static int FibRecursive(int Number)
    {
        Counter++;
        if (Number <= 1) return Number;
        else return FibRecursive(Number - 1) + FibRecursive(Number - 2);
    }
}
```

Таблица 1 – Код программы (наивный)

```
[Program] Введите число:
45
[Program] Результат: 1134903170
[Program] Количество выполнений: 3672623805
[Program] Время выполнения: 12.083369 сек.
```

Рисунок 1.1 – Результат выполнения программы и количество выполнений (наивный)

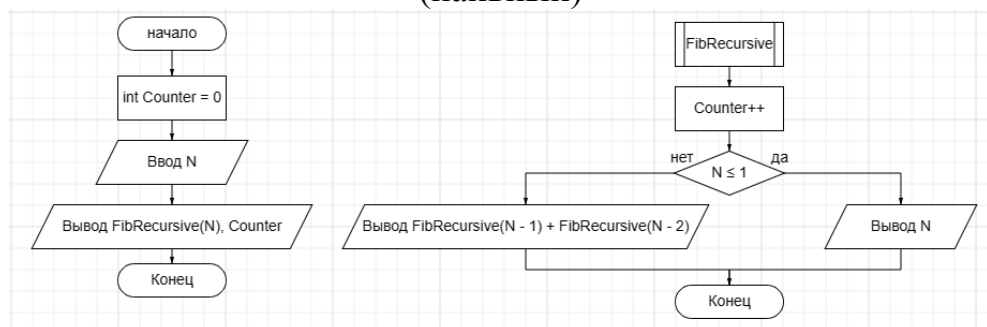


Рисунок 1.2 – Блок-схема программы (наивный)

```
using System;

public class HelloWorld
{
    public static int Counter = 0;
```

```

public static void Main(string[] args)
{
    Console.WriteLine("[Program] Введите число: ");
    int N = int.Parse(Console.ReadLine());
    var Timer = DateTime.Now;
    Console.WriteLine("[Program] Результат: " + FibArray(N).ToString());
    Console.WriteLine("[Program] Количество выполнений: " + Counter.ToString());
    Console.WriteLine("[Program] Время выполнения: " + DateTime.Now.Subtract(Timer).TotalSeconds + "
сек.");
}

static int FibArray(int Number)
{
    int[] Array = new int[Number + 1];
    Array[0] = 0;
    Array[1] = 1;
    for (int i = 2; i <= Number; i++) { Counter++; Array[i] = Array[i - 1] + Array[i - 2]; }
    return Array[Number];
}
}

```

Таблица 2 – Код программы (массив)

```

[Program] Введите число:
45
[Program] Результат: 1134903170
[Program] Количество выполнений: 44
[Program] Время выполнения: 0.003173 сек.

```

Рисунок 2.1 – Результат выполнения программы и количество выполнения (массив)

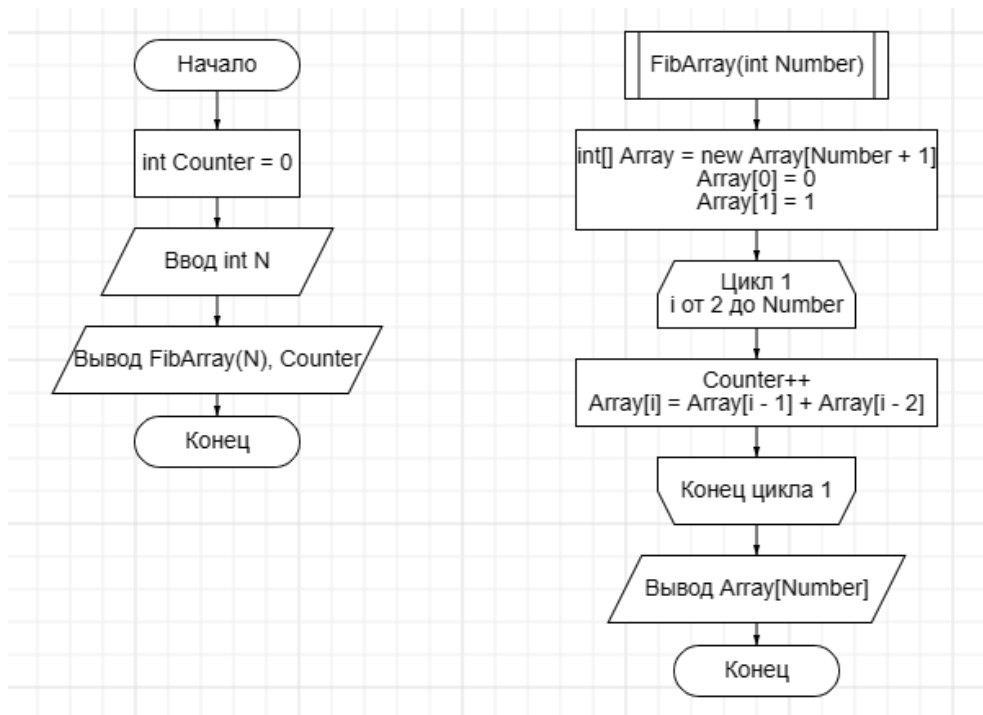


Рисунок 2.2 – Блок-схема программы (массив)

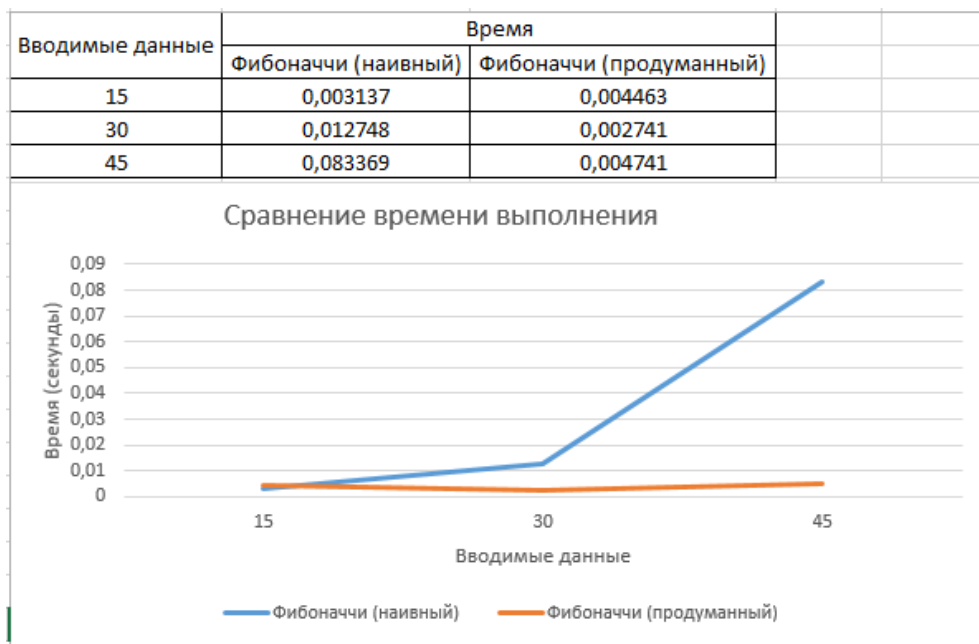


Рисунок 2.3 – Графики сравнений времени работы

```
using System;

class HelloWorld
{
    static int Counter = 0;

    static void Main()
    {
        Console.WriteLine("[System] Введите число A: ");
        int A = int.Parse(Console.ReadLine());
        Console.WriteLine("[System] Введите число B: ");
        int B = int.Parse(Console.ReadLine());
        var Timer = DateTime.Now;
        Console.WriteLine($"[System] НОД числа {A} и {B} = " + NaiveGCD(A, B).ToString());
        Console.WriteLine($"[system] Количество выполнений: {Counter}");
        Console.WriteLine("[System] Время выполнения: " +
DateTime.Now.Subtract(Timer).TotalSeconds.ToString());
    }

    static int NaiveGCD(int a, int b)
    {
        int GCD = 1;
        for (int d = 2; d < Math.Max(a, b); d++) { Counter++; if (a % d == 0 && b % d == 0) GCD = d; }
        return GCD;
    }
}
```

Таблица 3 – Код программы (наивный НОД)

```
[System] Введите число A:
7500955
[System] Введите число B:
12685
[System] НОД числа 7500955 и 12685 = 5
[System] Количество выполнений: 7500953
[System] Время выполнения: 0.056519
```

Рисунок 3.1 – Результат выполнения наивной программы (НОД)

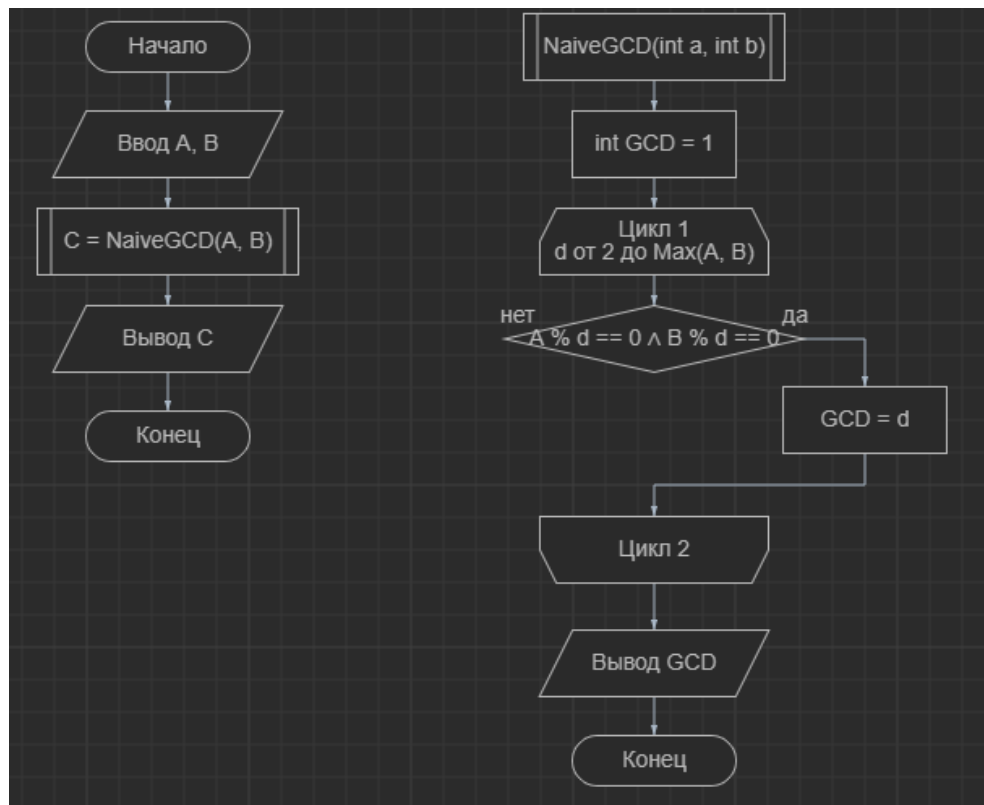


Рисунок 3.2 – Блок-схема наивной программы (НОД)

```
using System;

class HelloWorld
{
    static int Counter = 0;

    static void Main()
    {
        Console.WriteLine("[System] Введите число A: ");
        int A = int.Parse(Console.ReadLine());
        Console.WriteLine("[System] Введите число B: ");
        int B = int.Parse(Console.ReadLine());
```

```

var Timer = DateTime.Now;

Console.WriteLine($"[System] НОД числа {A} и {B} = " + GCD(A, B).ToString());
Console.WriteLine($"[System] Количество выполнения: {Counter}");
Console.WriteLine($"[System] Время выполнения: " +
DateTime.Now.Subtract(Timer).TotalSeconds.ToString());
}

static int GCD(int a, int b)
{
    if (a == 0) { Counter++; return b; }
    else if (b == 0) { Counter++; return a; }
    else if (a >= b) { Counter++; return GCD(a % b, b); }
    else { Counter++; return GCD(a, b % a); }
}
}

```

Таблица 4 – Код программы (продуманный НОД)

```

[System] Введите число A:
7500955
[System] Введите число B:
12685
[System] НОД числа 7500955 и 12685 = 5
[System] Количество выполнения: 8
[System] Время выполнения: 0.004314

```

Рисунок 4.1 – Результат выполнения продуманной программы (НОД)

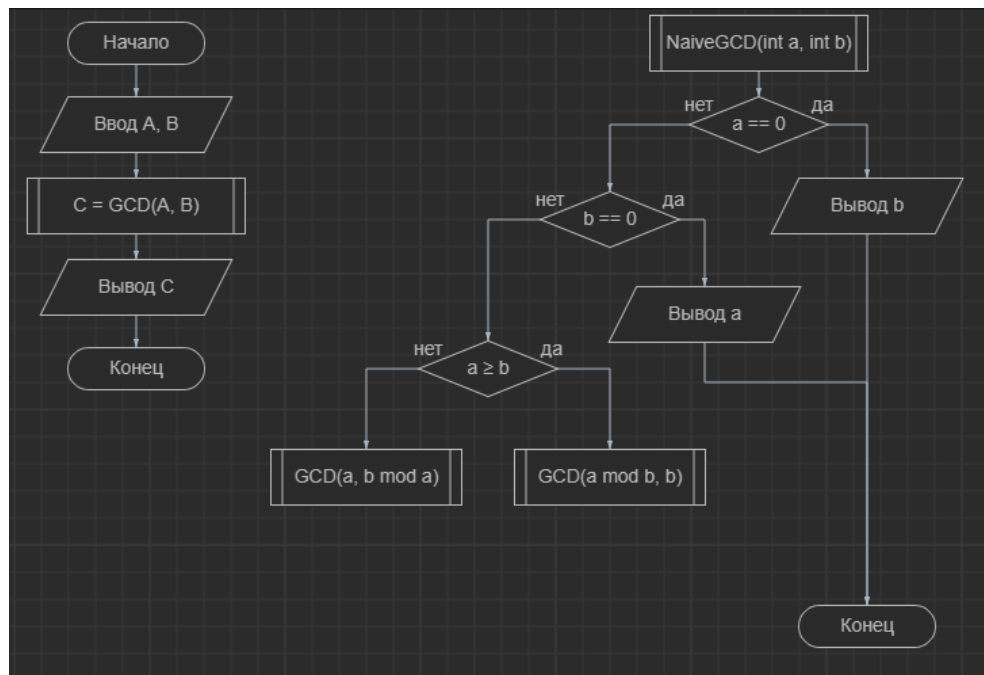


Рисунок 4.2 – Блок-схема продуманной программы (НОД)



Рисунок 4.3 – График сравнения времени работы

Вывод: В результате выполнения практической работы было проведено сравнение двух пар алгоритмов: наивные алгоритмы для поиска числа Фибоначчи и НОД, а также их продуманные версии. В результате было доказано, что разные реализации кода для решения одной и той же задачи могут существенно отличаться по эффективности.