

Лабораторная работа №1. Вспомогательные функции

Цель работы: приобретение навыков составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления.

Ход работы

1. Написание кода программы

Код файла Main.cpp представлен на рисунке 1.1.

```
#include "pch.h"
#include "Auxil.h"
#include <iostream>
#include <ctime>
#include <locale>

#define CYCLE 10000000

int Fibonacci(int i)
{
    int value = 0;
    if (i < 1) {
        return 0;
    }
    if (i == 1) {
        return 1;
    }
    return Fibonacci(i - 1) + Fibonacci(i - 2);
}

int _tmain(int argc, _TCHAR* argv[])
{
    double firstAverage = 0, secondAverage = 0;
    clock_t startTime = 0, endTime = 0;

    setlocale(LC_ALL, "rus");

    Auxil::start();
    startTime = clock();
    for (int i = 0; i < CYCLE; i++)
    {
        firstAverage += (double)Auxil::iget(-100, 100);
        secondAverage += Auxil::dget(-100, 100);
    }
    endTime = clock();

    cout << endl << "\nКоличество циклов: " << CYCLE;
    cout << endl << "Среднее значение (int): " << firstAverage / CYCLE;
    cout << endl << "Среднее значение (double): " << secondAverage / CYCLE;
    cout << endl << "Продолжительность (CLOCKS_PER_SEC): " << (endTime -
startTime);
    cout << endl << "Продолжительность (сек): " << ((double)(endTime -
startTime)) / ((double)CLOCKS_PER_SEC) << "\n";

    for (int n = 1; n < 51; n++)
    {
```

```

        startTime = clock();
        int num = Fibonacci(n);
        endTime = clock();
        cout << endl << n << "-е число Фибоначчи: " << num << ". Время на
нахождение числа: " << (endTime - startTime) << " CLOCKS_PER_SEC";
    }
    cout << endl;
    system("pause");

    return 0;
}

```

Рисунок 1.1 Содержание файла Main.cpp

Код файла Auxil.h представлен на рисунке 1.2.

```

#pragma once
#include <cstdlib>

namespace Auxil {
    void start();
    double dget(double rmin, double rmax);
    int iget(int rmin, int rmax);
}

```

Рисунок 1.2. Содержание файла Auxil.h

Код файла Auxil.cpp представлен на рисунке 1.3.

```

#include "pch.h"
#include "Auxil.h"

namespace Auxil
{
    void start()
    {
        srand((unsigned)time(NULL));
    };
    double dget(double rmin, double rmax)
    {
        return ((double)rand() / (double)RAND_MAX) * (rmax - rmin) + rmin;
    };
    int iget(int rmin, int rmax)
    {
        return (int)dget((double)rmin, (double)rmax);
    };
}

```

Рисунок 1.3. Содержание файла Auxil.cpp

Результат работы программы представлен на рисунке 1.4.

```

D:\Fourth_semestr\Math_Proc x + -
Количество циклов: 10000000
Среднее значение (int): 0.0112837
Среднее значение (double): 0.00587926
Продолжительность (CLOCKS_PER_SEC): 642
Продолжительность (сек): 0.642

1-е число Фибоначчи: 1. Время на нахождение числа: 0 CLOCKS_PER_SEC
2-е число Фибоначчи: 1. Время на нахождение числа: 0 CLOCKS_PER_SEC
3-е число Фибоначчи: 2. Время на нахождение числа: 0 CLOCKS_PER_SEC
4-е число Фибоначчи: 3. Время на нахождение числа: 0 CLOCKS_PER_SEC
5-е число Фибоначчи: 5. Время на нахождение числа: 0 CLOCKS_PER_SEC
6-е число Фибоначчи: 8. Время на нахождение числа: 0 CLOCKS_PER_SEC
7-е число Фибоначчи: 13. Время на нахождение числа: 0 CLOCKS_PER_SEC
8-е число Фибоначчи: 21. Время на нахождение числа: 0 CLOCKS_PER_SEC
9-е число Фибоначчи: 34. Время на нахождение числа: 0 CLOCKS_PER_SEC
10-е число Фибоначчи: 55. Время на нахождение числа: 0 CLOCKS_PER_SEC
11-е число Фибоначчи: 89. Время на нахождение числа: 0 CLOCKS_PER_SEC
12-е число Фибоначчи: 144. Время на нахождение числа: 0 CLOCKS_PER_SEC
13-е число Фибоначчи: 233. Время на нахождение числа: 0 CLOCKS_PER_SEC
14-е число Фибоначчи: 377. Время на нахождение числа: 0 CLOCKS_PER_SEC
15-е число Фибоначчи: 610. Время на нахождение числа: 0 CLOCKS_PER_SEC
16-е число Фибоначчи: 987. Время на нахождение числа: 0 CLOCKS_PER_SEC
17-е число Фибоначчи: 1597. Время на нахождение числа: 0 CLOCKS_PER_SEC
18-е число Фибоначчи: 2584. Время на нахождение числа: 0 CLOCKS_PER_SEC
19-е число Фибоначчи: 4181. Время на нахождение числа: 0 CLOCKS_PER_SEC
20-е число Фибоначчи: 6765. Время на нахождение числа: 0 CLOCKS_PER_SEC
21-е число Фибоначчи: 10946. Время на нахождение числа: 0 CLOCKS_PER_SEC
22-е число Фибоначчи: 17711. Время на нахождение числа: 0 CLOCKS_PER_SEC

```

Рисунок 1.4. Результат работы программы

2. Измерение скорости выполнения функции генерации случайных чисел.

Результаты измерений и соответствующий график представлен на рисунке 2.1.

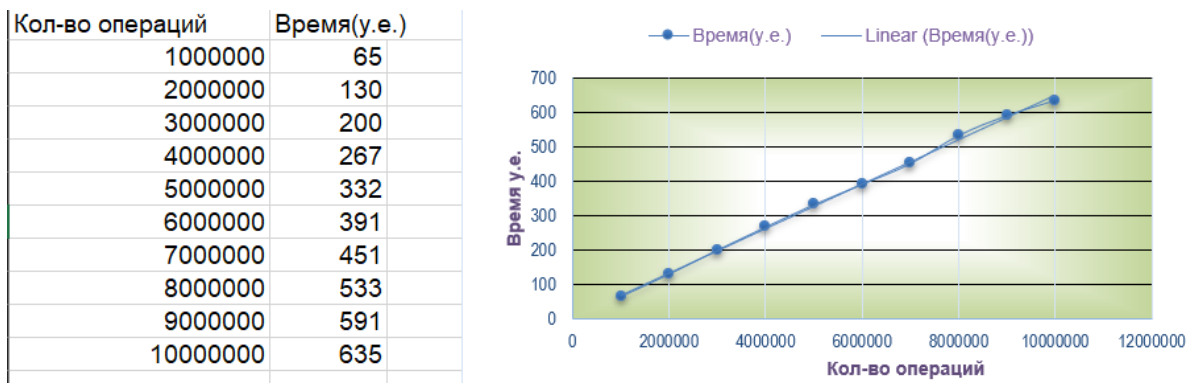


Рисунок 2.1. Результаты измерений и их график

Вывод: скорость выполнения программы линейно зависит от количества итераций цикла.

3. Измерение скорости выполнения функции подсчёта n-ого числа Фибоначчи.

Результаты измерений и соответствующий график приведены на рисунке 3.1.

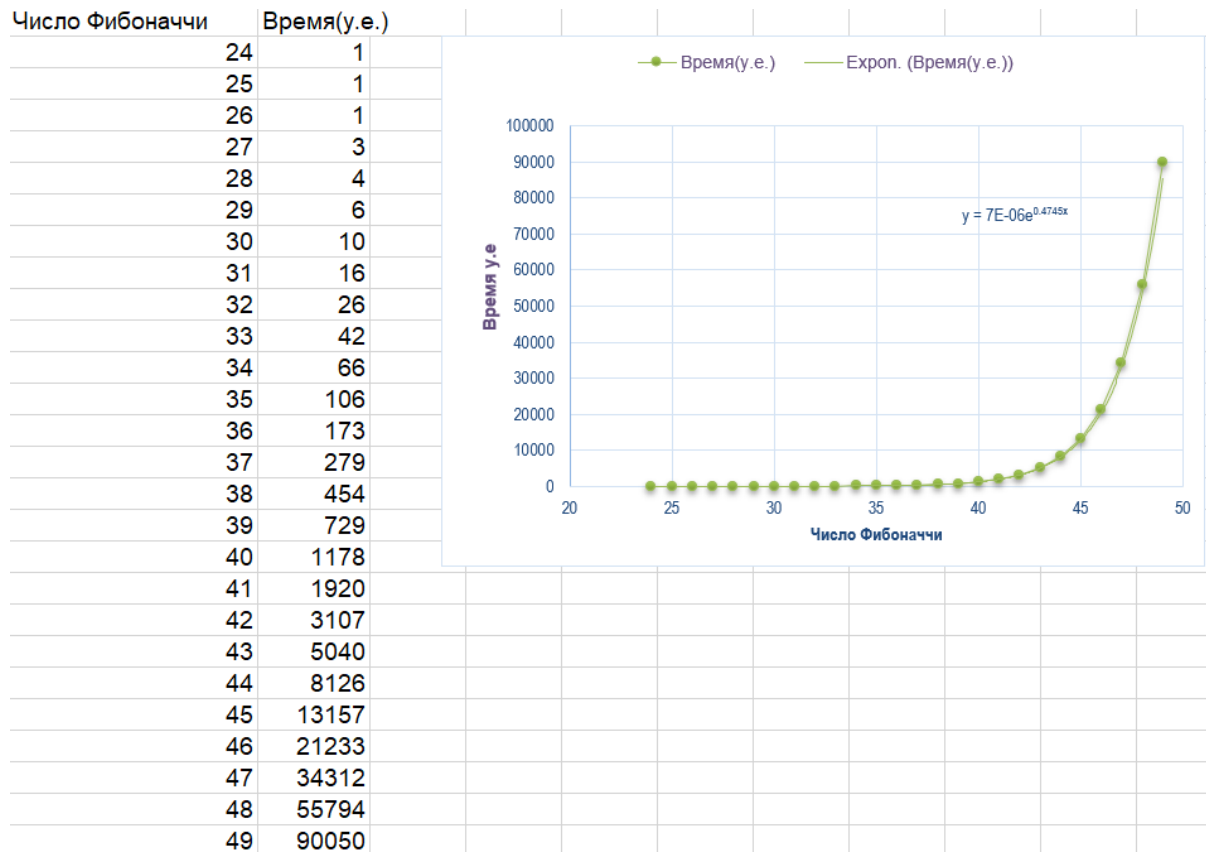


Рисунок 3.1. Результаты измерений и их график

Вывод: скорость выполнения программы экспоненциально зависит от порядкового номера числа Фибоначчи.