

Лабораторная работа №1. Вспомогательные функции

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

Ход работы

1.1. Написание кода программы.

Код файла Lab1.cpp приведен в листинге 1.1.

```
#include "stdafx.h"
#include "Auxil.h" // вспомогательные функции
#include <iostream>
#include <ctime>
#include <locale>

using namespace std;
int fibNumber(unsigned int);

#define CYCLE 1000000 // количество циклов
int main()
{
    double av1 = 0, av2 = 0;
    clock_t t1 = 0, t2 = 0;
    setlocale(LC_ALL, "rus");
    auxil::start(); // старт генерации
    t1 = clock(); // фиксация времени
    for (int i = 0; i < CYCLE; i++)
    {
        av1 += (double)auxil::iget(-100, 100); // сумма случайных чисел
        av2 += auxil::dget(-100, 100); // сумма случайных чисел
    }
    t2 = clock(); // фиксация времени
    std::cout << std::endl << "Количество циклов: " << CYCLE;
    std::cout << std::endl << "Среднее значение (int): " << av1 / CYCLE;
    std::cout << std::endl << "Среднее значение (double): " << av2 / CYCLE;
    std::cout << std::endl << "Продолжительность (y.e): " << (t2 - t1);
    std::cout << std::endl << " (сек): " << ((double)(t2 - t1)) / ((double)CLOCKS_PER_SEC);
    std::cout << std::endl;
    //Рекурсия
    for (int i = 20; i < 45; i++)
    {
        double startTime2 = clock();
        cout << i << "-ое число Фибоначчи: " << fibNumber(i) << endl;
        double endTime2 = clock();
        double resultTime2 = (endTime2 - startTime2) / CLOCKS_PER_SEC;
        cout << "Продолжительность (y.e): " << (endTime2 - startTime2) << endl;
    }
    system("pause");
    return 0;
}
```

Листинг 1.1. Содержание файла main.cpp

Код файла Auxil.h приведен в листинге 1.2.

```
#pragma once
#include <cstdlib>
namespace auxil
{
    void start(); // старт генератора сл. чисел
    double dget(double rmin, double rmax); // получить случайное число
    int iget(int rmin, int rmax); // получить случайное число
};
```

Листинг 1.2. Содержание файла Auxil.h

Код файла Auxil.cpp приведен в листинге 1.3.

```
#include "stdafx.h"
#include "Auxil.h"
#include <ctime>
namespace auxil
{
    void start() // старт генератора сл. чисел
    {
        srand((unsigned)time(NULL));
    };
    double dget(double rmin, double rmax) // получить случайное число
    {
        return ((double)rand() / (double)RAND_MAX) * (rmax - rmin) + rmin;
    };
    int iget(int rmin, int rmax) // получить случайное число
    {
        return (int)dget((double)rmin, (double)rmax);
    };
};
```

Листинг 1.3. Содержание файла Auxil.cpp

1.2. Измерение скорости выполнения функции генерации случайных чисел.

Результаты измерений и соответствующий график приведены на рисунке 1.1

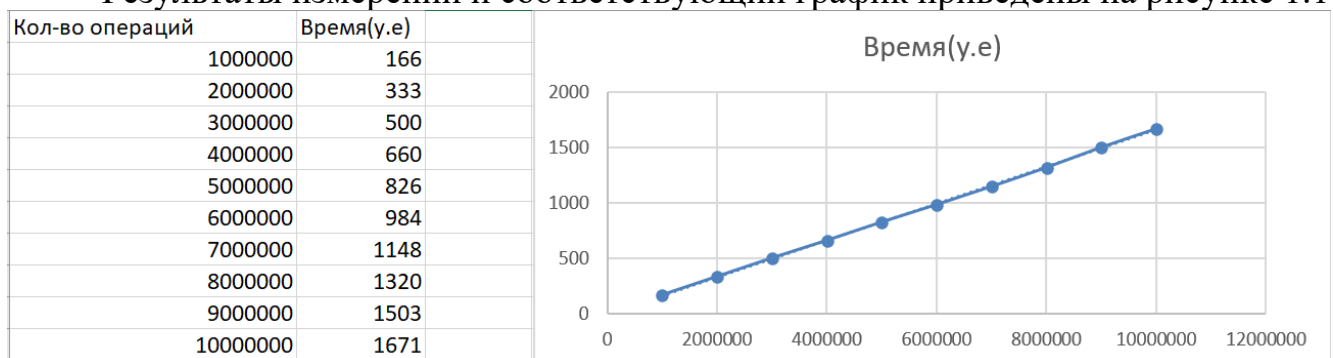


Рисунок 1.1. Результаты измерений и их график

Результат работы программы представлен на рисунке 1.2

```

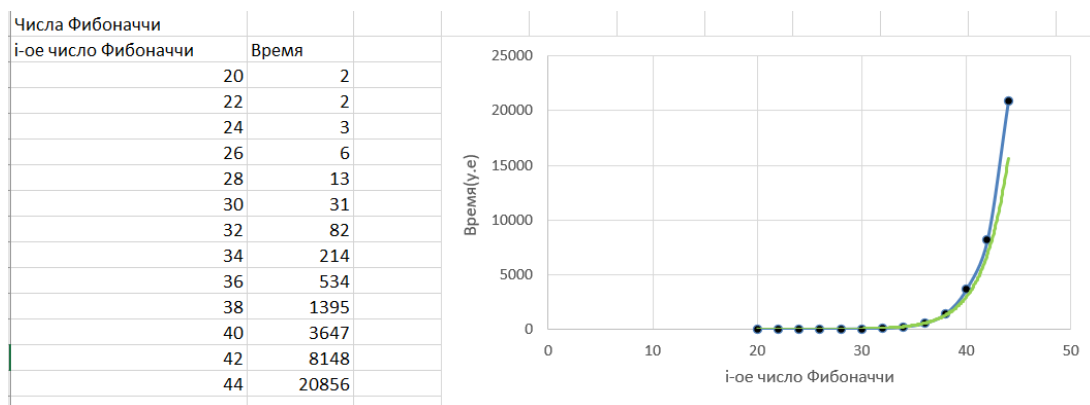
Количество циклов:      10000000
Среднее значение (int):  -0.0172659
Среднее значение (double): 0.00744104
Продолжительность (y.e): 1683
                        (сек): 1.683

```

Рисунок 1.2 Результат работы программы

1.3. Измерение скорости выполнения функции подсчета чисел Фибоначчи.

Результаты измерений и соответствующий график приведены на рисунке 1.3



Результат работы программы представлен на рисунке 1.4

```

20-ое число Фибоначчи: 6765
Продолжительность (y.e): 2
22-ое число Фибоначчи: 17711
Продолжительность (y.e): 2
24-ое число Фибоначчи: 46368
Продолжительность (y.e): 3
26-ое число Фибоначчи: 121393
Продолжительность (y.e): 6
28-ое число Фибоначчи: 317811
Продолжительность (y.e): 13
30-ое число Фибоначчи: 832040
Продолжительность (y.e): 31
32-ое число Фибоначчи: 2178309
Продолжительность (y.e): 82
34-ое число Фибоначчи: 5702887
Продолжительность (y.e): 214
36-ое число Фибоначчи: 14930352
Продолжительность (y.e): 534
38-ое число Фибоначчи: 39088169
Продолжительность (y.e): 1395
40-ое число Фибоначчи: 102334155
Продолжительность (y.e): 3647
42-ое число Фибоначчи: 267914296
Продолжительность (y.e): 8148
44-ое число Фибоначчи: 701408733
Продолжительность (y.e): 20856

```

Рисунок 1.4 Результат выполнения программы

Вывод: скорость выполнения функции генерации случайных чисел линейно зависит от количества итераций цикла. Скорость выполнения программы по расчету i -ого числа Фибоначчи зависит экспоненциально.