**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 main.cpp приведен в листинге 1.1.

|  |
| --- |
| #include "stdafx.h"  #include "Auxil.h" // вспомогательные функции  #include <iostream>  #include <ctime>  #include <locale>  #define CYCLE 1000000 // количество циклов  int Fib(int i)  {  int value = 0;  if (i < 1)  return 0;  if (i == 1)  return 1;  return Fib(i - 1) + Fib(i - 2);  }  int \_tmain(int argc, \_TCHAR\* argv[])  {  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 (у.е): " << (t2 - t1);  std::cout << std::endl << " (сек): " << ((double)(t2 - t1)) / ((double)CLOCKS\_PER\_SEC);  std::cout << std::endl;  for (int n = 20; n < 41; n++)  {  t1 = clock();  int num = Fib(n);  t2 = clock();  std::cout << std::endl << n << "-е число Фибоначчи " << t2 - t1 << " у.е.";  }  std::cout << std::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  }; |

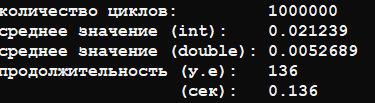
Листинг 1.2. Содержание файла Auxil.h

Код файла Auxil.cpp приведет 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

Результат работы программы представлен на рисунке 1.1.



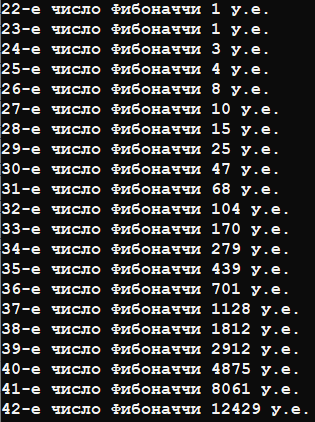


Рисунок 1.1 Результаты работы программы

## **1.2. Измерение скорости выполнения функции генерации случайных чисел.**

Результаты измерений и соответствующий график приведены на рисунке 1.2.

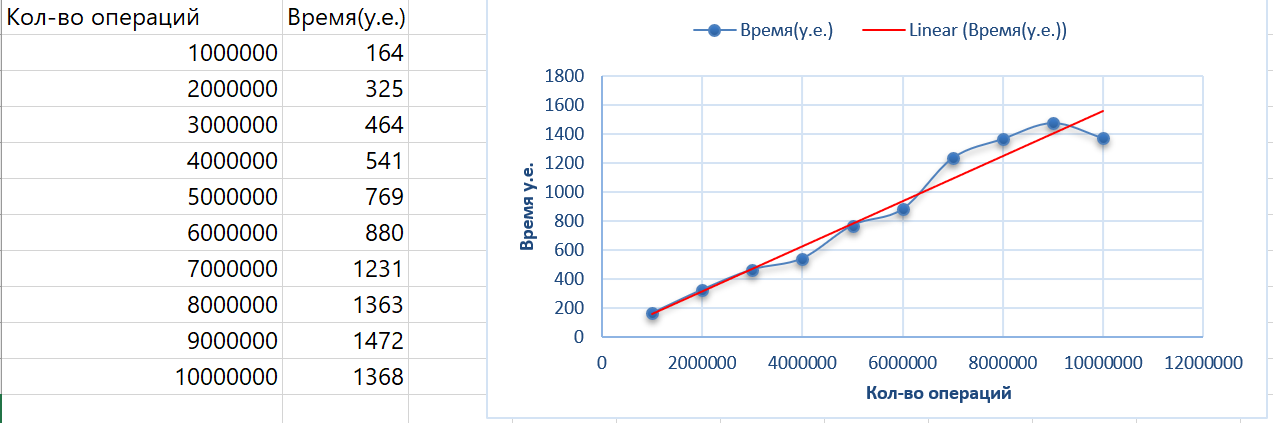


Рисунок 1.2. Результаты измерений и их график

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

## **1.3. Измерение скорости выполнения функции подсчета n-го числа Фибоначчи.**

Результаты измерений и соответствующий график приведены на рисунке 1.3.

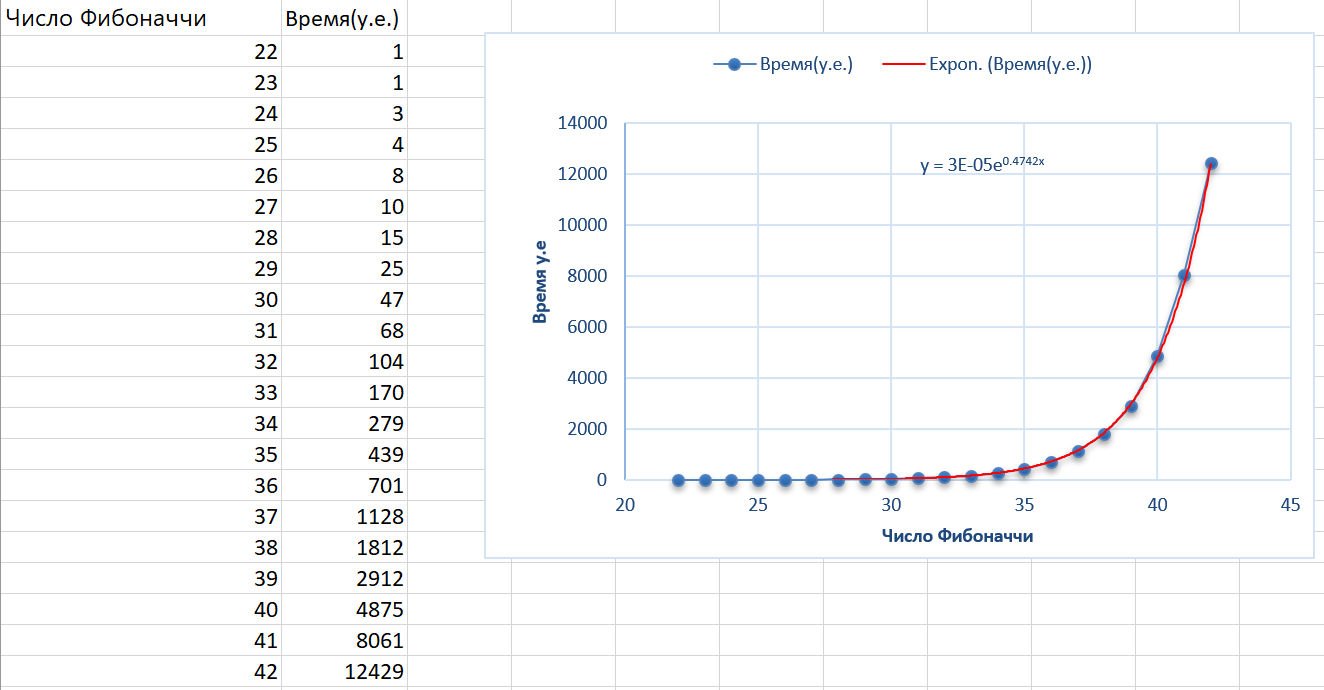


Рисунок 1.3. Результаты измерений и их график

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