**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.

Функция rec(int num) находит число Фибоначчи с порядковым номером n,

Используя рекурсивный алгоритм

Auxil() – функция, задающая старт генерации случайных чисел в программе

После старта генерации случайных чисел следует функция clock(), которая возвращает время работы программы в тиках(единица измерения процессорного времени)

Между двумя функциями clock() находится цикл, в котором высчитываются суммы случайных чисел, в диапазоне от -100 до 100

После цикла начинается вычисление числа Фибоначчи с порядковым номером num (rec(num)), при этом вычисление происходит между вызовами функции clock();

Код файла Auxil.h приведен в листинге 1.2.

Файл Auxil.h содержит прототипы функций

start(),

dget(double rmin, double rmax),

iget(int rmin, int rmax)

Код файла Auxil.cpp приведет в листинге 1.3.

Auxil.cpp содержит пространство имен auxil, в котором содержится 3 функции:

start(),

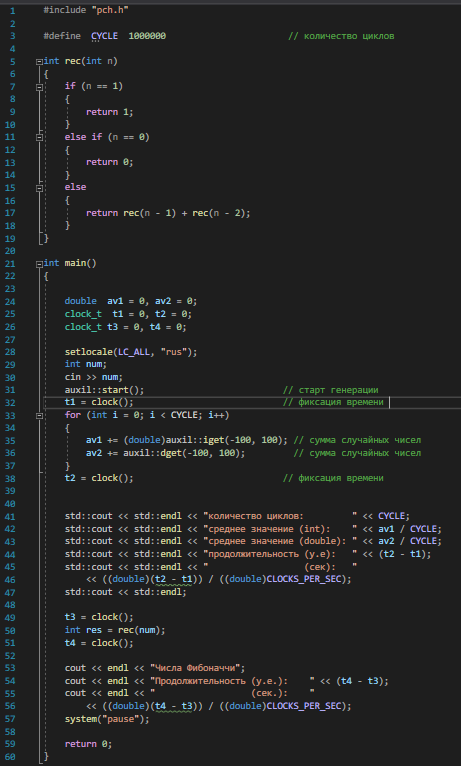
dget(double rmin, double rmax),

iget(int rmin, int rmax)

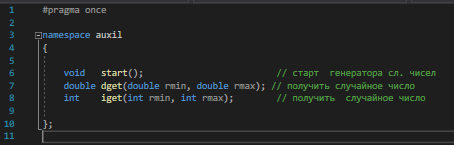
функция start() необходима для старта генерации псевдослучайных чисел

функция dget(double rmin, double rmax) генерирует псевдослучайное число в диапазоне от rmin до rmax типа double.

Функция iget(int rmin, int rmax) генерирует псевдослучайное число в диапазоне от rmin до rmax типа int.



Листинг 1.1. Содержание файла main.cpp



Листинг 1.2. Содержание файла Auxil.h

Код файла Auxil.cpp приведет в листинге 1.3.



Листинг 1.3. Содержание файла Auxil.cpp

Результат работы представлен на рисунке 1.1.

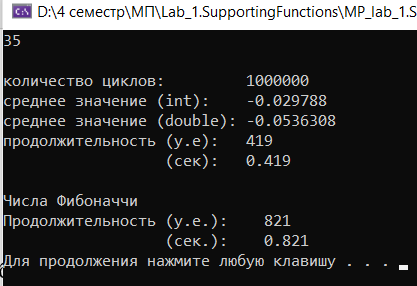


Рисунок 1.1. Результат работы кода

## **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.

Также был проверен алгоритм нахождения n числа Фибоначчи.

## **1.3. Измерение скорости вычисления n числа.**

Результаты измерений и соответствующий график приведены на рисунке 1.3

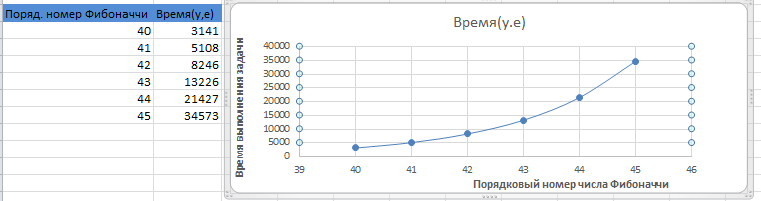


Рисунок 1.3 Результаты измерений и их график

Вывод: сложность данного алгоритма является экспоненциальной. Это подтверждает и график.