Белорусский государственный технологический университет

факультет информационных технологий

кафедра информационных систем и технологий

**Отчёт по предмету**

**«Математическое программирование»**

Немкович Анастасия Вадимовна

ФИТ, 2 курс, группа 1/1

Минск 2023

**Лабораторная №1. Вспомогательные функции**

**ЦЕЛЬ РАБОТЫ:** приобретение навыков составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления.

***Задание 1.*** Разработайте три функции (start, dget и iget) используя следующие спецификации:

|  |
| --- |
| **void start();**  // функция устанавливает в качестве начального числа для //генератора псевдослучайных чисел текущее значение // системного времени в фомате функции time() |

|  |
| --- |
| **double dget(**  **double rmin,** //[in]минимальное значение  **double rmax** //[in]максимальное значение  **);**  //-- функция возвращает действительное псевдослучайное число в // диапазоне оn **rmin** до **rmax** |

|  |
| --- |
| **int iget(**  **int rmin,** //[in]минимальное значение  **int rmax** //[in]максимальное значение  **);**    //-- функция возвращает целое псевдослучайное число в  // диапазоне оn **rmin** до **rmax** |

Программный код:

|  |
| --- |
| //-- Auxil.h  #pragma once  #include "pch.h"  #include <cstdlib>  #include <ctime>  namespace auxil  {  void start(); // старт генератора случайных чисел  double dget(double rmin, double rmax); // получить действительное случайное число  int iget(int rmin, int rmax); // получить целое случайное число  } |

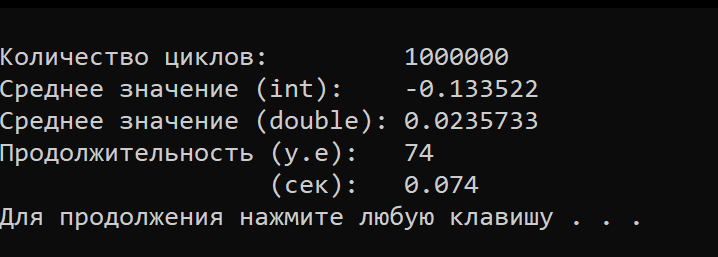
|  |
| --- |
| //-- Auxil.cpp  #include "pch.h"  #include "Auxil.h"  namespace auxil  {  void start() // старт генератора случайных чисел  {  srand((unsigned)time(NULL));  };  double dget(double rmin, double rmax) // получить действительное случайное число  {  return ((double)rand() / (double)RAND\_MAX) \* (rmax - rmin) + rmin;  };  int iget(int rmin, int rmax) // получить целое случайное число  {  return (int)dget((double)rmin, (double)rmax);  };  } |

***Задание 2***

1. Реализовать пример 2.
2. Для проверки работоспособности разработанных функций и приобретения навыков замера продолжительности процесса вычисления реализуйте программу, приведенную в примере 2.

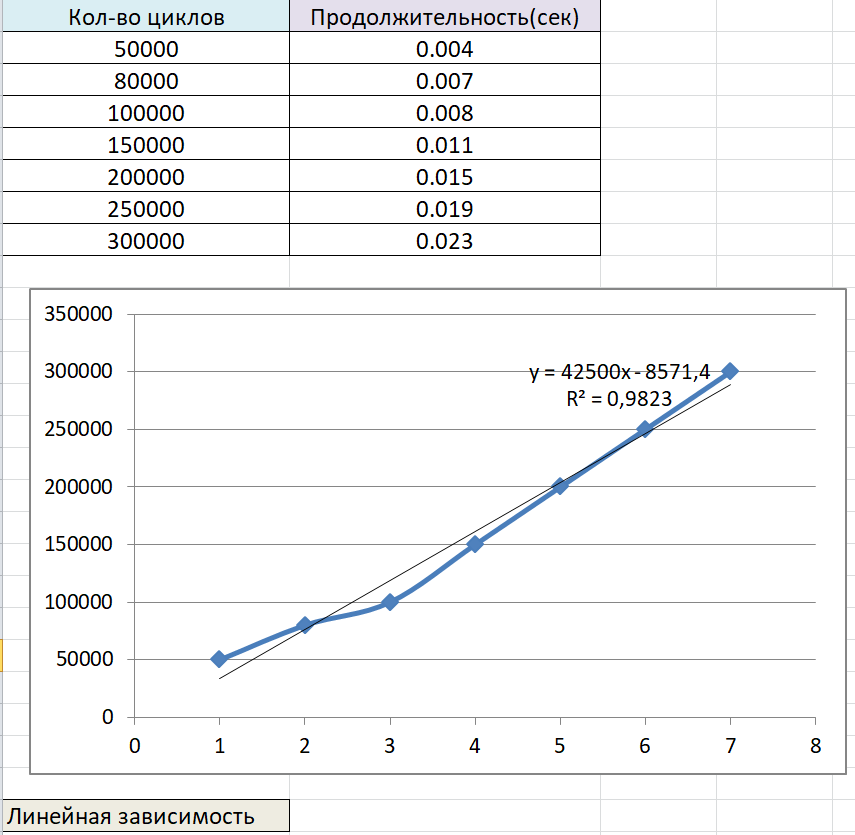
|  |
| --- |
| //-- 1lab.cpp  #include "pch.h"  #include "Auxil.h"  #include <iostream>  #include <ctime>  #include <locale>  #define CYCLE 1000000 // количество циклов  int main()  {  double av1 = 0, av2 = 0;  clock\_t t1 = 0, t2 = 0;  setlocale(LC\_ALL, "rus");  auxil::start(); // старт генерации  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 (у.е): " << (t2 - t1);  std::cout << std::endl << " (сек): "  << ((double)(t2 - t1)) / ((double)CLOCKS\_PER\_SEC);  std::cout << std::endl;  system("pause");  return 0; |

Результат выполнения:



**Задание 3**

Проведите необходимые эксперименты и постройте график зависимости (Excel) продолжительности процесса вычисления от количества циклов в примере 2. Проанализируйте характер зависимости.



Проведите исследование любого другого итерационного алгоритма, например, вычисления факториала или генератора чисел Фибоначчи (прим. – например вычислите каким будет 100-е, 200-е, 300-е и т.д число), и включите в отчет график.

|  |
| --- |
| cout << "3 задание" << endl << endl;  auxil::start(); // Инициализация генератора случайных чисел  int n = 50;  cout << "Генерация последовательности чисел Фибоначчи до " << n << " члена:\n";  clock\_t start = clock(); // Запуск таймера  for (int i = 0; i < n; ++i)  {  int fib = auxil::fibonacci(i);  cout << fib << " ";  }  clock\_t end = clock(); // Остановка таймера  cout << std::endl;  double time\_taken = double(end - start) / double(CLOCKS\_PER\_SEC);  cout << "Время выполнения программы: " << time\_taken << " секунд" << endl;  return 0; |

