## **Лабораторная работа 1. Вспомогательные функции**

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

Проверка работоспособности трех функций: start, dget и iget и приобретения навыков замера продолжительности процесса вычисления.

|  |
| --- |
| #include <stdio.h>  #include "Auxil.h" // вспомогательные функции  #include <iostream>  #include <ctime>  #include <locale>  #include "Factorial.h"  #define CYCLE 2000000 // количество циклов  using namespace std;  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  cout << endl << "количество циклов: " << CYCLE;  cout << endl << "среднее значение (int): " << av1 / CYCLE;  cout << endl << "среднее значение (double): " << av2 / CYCLE;  cout << endl << "продолжительность (у.е): " << (t2 - t1);  cout << endl << " (сек): " << ((double)(t2 - t1)) / ((double)CLOCKS\_PER\_SEC);  cout << endl;  clock\_t t3 = 0, t4 = 0;  t3 = clock();  long long res = fact::factorial(20); // 20 - предел  t4 = clock();  cout << "\nрезультат: " << res;  cout << endl << "продолжительность (у.е): " << (t4 - t3);  cout << endl << " (сек): " << ((double)(t4 - t3)) / ((double)CLOCKS\_PER\_SEC);  cout << endl;  return 0;  } |

Код главного файла программы

Код представляет программу, которая выполняет вычисления и измеряет время их выполнения. В программе используются функции для генерации случайных чисел и вычисления факториала. Она также выводит на консоль средние значения случайных чисел и время выполнения для каждой части кода.

|  |
| --- |
| //-- Auxil.cpp  #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);  };  } |
| #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  }; |

Код файлов Auxil.cpp и Auxil.h

Данный код представляет собой реализацию набора функций в пространстве имен auxil, которые используются для генерации случайных чисел. Функция start() инициализирует генератор случайных чисел, используя текущее время в качестве семени. Функция dget() возвращает случайное число типа double в заданном диапазоне, а функция iget() возвращает случайное целое число типа int в заданном диапазоне.

|  |
| --- |
| #include "Factorial.h"  #include <iostream>  using namespace std;  namespace fact {  long double factorial(int num)  {  if (num == 1) return 1;  return num \* factorial(num - 1);  }  } |
| #pragma once  namespace fact {  long double factorial(int num);  } |

Код файлов Factorial.cpp и Factorial.h

Код представляет собой реализацию вычислений факториала. Функция factorial() рекурсивно вычисляет факториал числа num. Если num равно 1, функция возвращает 1. В противном случае, функция вызывает саму себя с аргументом num - 1 и умножает результат на num, возвращая произведение.

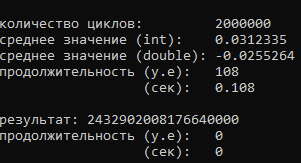


Рисунок 4 – Результат выполнения программы в консоли

**Графики продолжительности вычислений в условных единицах процессорного времени**

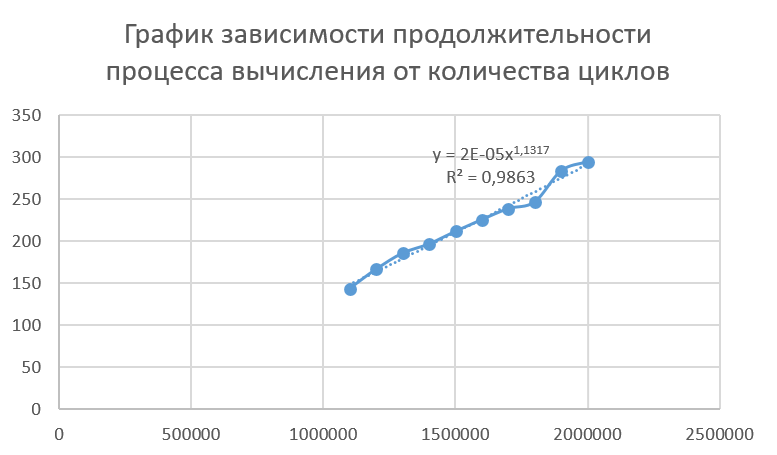
****

Рисунок 1 - График зависимости продолжительности процесса вычисления от количества циклов(степенная)

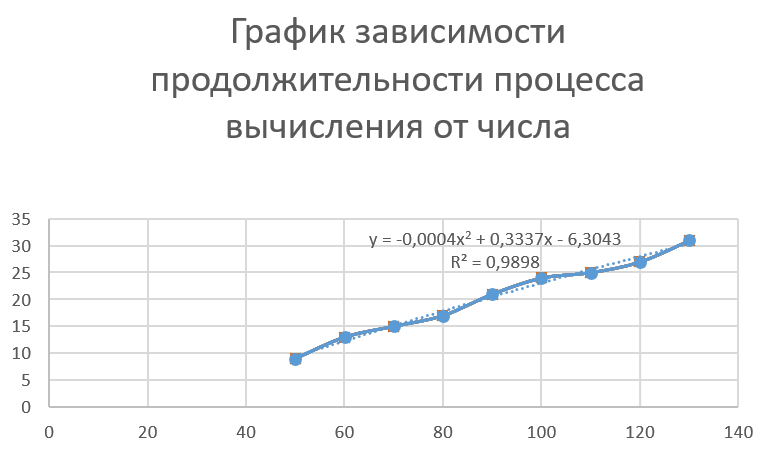
****

Рисунок 2 – График зависимости продолжительности процесса вычисления от количества циклов(полиномиальная)

Вывод по лабораторной работе: степенная зависимость на первом графике и полиномиальная на втором, графики увеличиваются, значит увеличение количества циклов влияет на продолжительность процесса. Зависимости представлены на графиках. В ходе выполнения лабораторной работы была изучена зависимость продолжительности процесса от количества циклов. Приобретены навыки составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления.