

Учреждение образования
«Белорусский государственный технологический университет»

Лабораторная работа №1

Вспомогательные функции

Выполнила:

Студентка 2 курса 1 группы ФИТ

Самсоник Анастасия Ивановна

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

Заголовочный файл Auxil.h

```
#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
};
```

Файл Auxil.cpp с разработанными тремя функциями (start, dget и iget)

```
#include "Auxil.h"
#include <ctime>
namespace auxil
{
    void start() // старт генератора сл. чисел
    {
        srand((unsigned)time(NULL));
    };
    double dget(double rmin, double rmax) // функция возвращает действительное
    псевдослучайное число в диапазоне от rmin до rmax
    {
        return ((double)rand() / (double)RAND_MAX) * (rmax - rmin) + rmin;
    };
    int iget(int rmin, int rmax) // функция возвращает целое псевдослучайное
    число в диапазоне от rmin до rmax
    {
        return (int)dget((double)rmin, (double)rmax);
    };
}
```

Файл lab_1.cpp для проверки работоспособности разработанных функций

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

#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;
system("pause");
return 0;
}

```

```

C:\Users\Acer\Desktop\лабы\матпрог\lab_1\x64\Debug\lab_1.exe

количество циклов:      1000000
среднее значение (int):  0.035098
среднее значение (double): -0.0391687
продолжительность (y.e): 252
                        (сек): 0.252
Для продолжения нажмите любую клавишу . . .

```

Далее код был изменен, чтобы усложнить выполнения цикла

```

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

#define CYCLE 600000 // количество циклов
double f(double x)
{
    return sin(x);
}
int main()
{
    double av1 = 0, av2 = 0;
    clock_t t1 = 0, t2 = 0;
    double Integral = 0;
    double a = 0.0, b = 1.0;
    double h = 0.1;
    double n; // задаём число разбиений n

    n = (b - a) / h;
    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
        Integral += h * f(a + h * (i - 0.5));
    }
    t2 = clock(); // фиксация времени
}

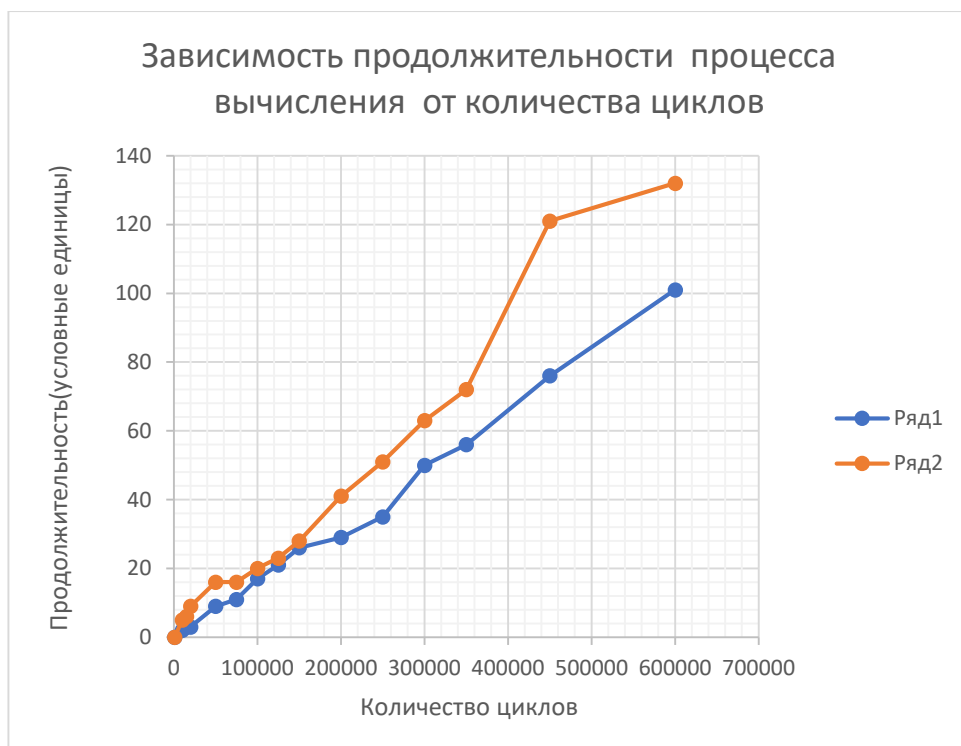
```

```

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;
system("pause");
return 0;
}

```

График зависимости (Excel) продолжительности процесса вычисления от количества циклов



ВЫВОД: В ходе выполнения лабораторной работы была исследована зависимость продолжительности процесса вычисления от количества циклов. Были приобретены навыки составления и отладки программ с использованием пользовательских функций для замера продолжительности

процесса вычисления. График зависимости продолжительности процесса вычисления от количества циклов показан на примерах двух экспериментов и является линейным.