МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №4

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Операторы цикла: вычисления с заданной точностью»

Работу выполнил

студент гр. 4141 В.С. Сыворотнев

Санкт-Петербург

2022

***Цель лабораторной работы:*** *изучение концепций и освоение технологии процедурного программирования, приобретение навыков процедурного программирования на языке C/C++ циклических вычислений.*

***Задание на программирование:*** *используя технологию процедурного программирования разработать программу решения индивидуальной задачи тремя видами циклических управляющих структур: Цикл - Пока (с предусловием), Цикл - До (с постусловием), Цикл - Для (с параметром).*

***Вариант № 31:***

Вычислить предел последовательности {*Yn*} при *n*→, где *Yn* вычисляется по формуле:



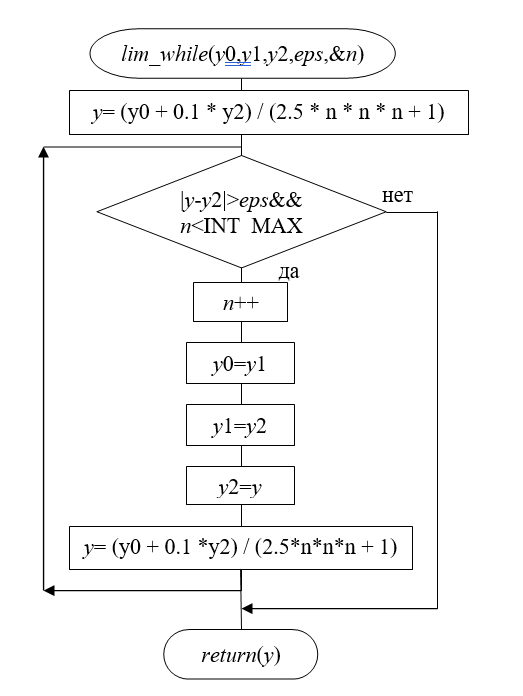
Значения *Y*0, *Y*1, *Y*2 и точность вычисления *ε* вводятся с клавиатуры. Вычисления прекратить при выполнении условия ⎪*Yn*-*Yn*-1⎪≤*ε*.

***а) Математическая модель решения с использованием цикла while***

При вызове в функцию lim\_while передаются значения *ε*, *Y*0, *Y*1, *Y*2 и указатель на параметр *n*. На основе этих значений для значения параметра *n*=3 вычисляется значение *Y* по заданной рекуррентной формуле. Проверяется выполнение условия прекращения вычислений ⎪*Y*–*Y*2⎪≤*ε*. Для контроля числа выполненных итераций одновременно проверяется выполнение условия *n*<INT\_MAX.

Если ⎪*Y*–*Y*2⎪>*ε*, для вычисления следующего значения *Y* увеличивается на 1 значение параметра *n*, значение *Y*1 переписывается в *Y*0, значение *Y*2 переписывается в *Y*1, значение *Y* переписывается в *Y*2 и снова вычисляется значение *Y*. Действия повторяются до выполнения условия ⎪*Y*–*Y*2⎪≤*ε*. Необходимо обратить внимание, что очередное значение *Y* **всегда вычисляется непосредственно перед проверкой** выполнения условия прекращения вычислений. При завершении работы функция возвращает значение *Y*. Значение параметра *n* возвращается через указатель.

*Схема алгоритма решения с использованием цикла* ***while (****функция* ***lim\_while)***

******

***б) Математическая модель решения с использованием цикла do…while***

Необходимо обратить внимание, что поскольку очередное значение *Y* всегда вычисляется **непосредственно перед проверкой** выполнения условия прекращения вычислений, построение схемы алгоритма с использованием цикла do…while **целесообразно вести** **снизу вверх** от проверки выполнения условия прекращения вычислений ⎪*Y*–*Y*2⎪≤*ε*.

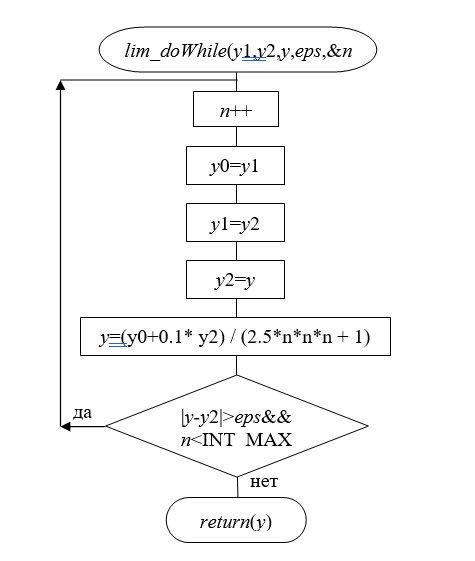
Перед этой проверкой вычисляется очередное значение *Y* по заданной рекуррентной формуле. Перед этим предыдущее значение *Y* переписывается в *Y*2, перед этим значение *Y*2 переписывается в *Y*1, перед этим значение *Y*1 переписывается в *Y*0, перед этим увеличивается на 1 значение параметра *n*.

Поэтому для правильного выполнения цикла **параметр *n* при вызове функции** (т.е. перед входом в цикл) **принимает значение 2**, а в функцию должны быть переданы значения *ε*, *Y*0, *Y*1, *Y*2 и указатель на параметр *n*. При вызове функции значения *Y*0, *Y*1, *Y*2 присваиваются соответственно переменным *Y*1, *Y*2, *Y*.

Действия повторяются до выполнения условия ⎪*Y*–*Y*2⎪≤*ε*. Для контроля числа выполненных итераций проверяется выполнение условия *n*<INT\_MAX.

При завершении работы функция возвращает значение *Y*. Значение параметра *n* возвращается через указатель.

*Схема алгоритма решения с использованием цикла* ***do…while (****функция* ***lim\_doWhile)***

******

***в) Математическая модель решения с использованием цикла for***

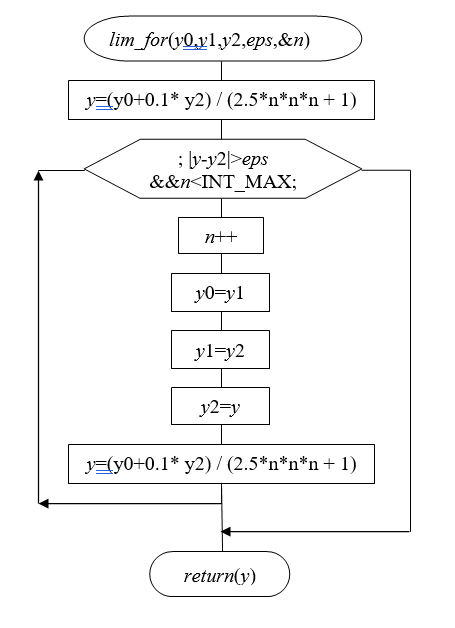
При вызове в функцию ***predel\_*3** передаются значения *ε*, *Y*0, *Y*1, *Y*2 и указатель на параметр *n*. На основе этих значений для начального значения параметра *n*=3 вычисляется значение *Y* по заданной рекуррентной формуле. Проверяется выполнение условия прекращения вычислений ⎪*Y*–*Y*2⎪≤*ε*. Для контроля числа выполненных итераций одновременно проверяется выполнение условия *n*<INT\_MAX.

Если ⎪*Y*–*Y*2⎪>*ε*, для вычисления следующего значения *Y* увеличивается на 1 значение параметра *n*, значение *Y*1 переписывается в *Y*0, значение *Y*2 переписывается в *Y*1, значение *Y* переписывается в *Y*2 и снова вычисляется значение *Y*.

Действия повторяются до выполнения условия ⎪*Y*–*Y*2⎪≤*ε*. Необходимо обратить внимание, что очередное значение *Y* всегда вычисляется **непосредственно перед проверкой** выполнения условия прекращения вычислений.

При завершении работы функция возвращает значение *Y*. Значение параметра *n* возвращается через указатель.

*Схема алгоритма решения с использованием цикла* ***for (****функция* ***lim\_for)***

******

***Текст программы***

#include<iostream>

#include<cmath>

#include<limits.h>

#include<locale.h>

using namespace std;

double lim\_while(double, double, double, double, int& n);

double lim\_doWhile(double, double, double, double, int& n);

double lim\_for(double, double, double, double, int& n);

int main() {

    int key,

    n1, n2, n3;             //число итераций для разных циклов

    double first\_ans, second\_ans, third\_ans,    //результаты расчетов

    y,                      //значение предела

    y0, y1, y2;             //текущие значения элементов последовательности

    double eps;             //точность вычисления предела

    setlocale(LC\_ALL, "Russian");

    for (;;) {

        //Ввод исходных данных

        cout << " Вид действия:" << endl;

        cout << " 1 - вычисление предела последовательности" << endl;

        cout << " 2 - завершение задачи" << endl;

        cout << " Введите вид действия -> ";

        cin >> key;

        switch (key) {

        case 1:

            //Ввод исходных данных

            cout << " Введите значения y0, y1, y2 -> ";

            cin >> y0 >> y1 >> y2;

            cout << " Введите точность вычисления -> ";

            cin >> eps;

            if (eps <= 0 || eps > .1) {

                cout << "Ошибка ввода. Значение eps д.б. > 0 и  <=0.1" << endl;

                continue;

            }

            //Решение циклом while

            first\_ans = lim\_while(y0, y1, y2, eps, n1 = 3);

            if (n1 != INT\_MAX) {

                cout.precision(4);  //число знаков после десятичной точки

                cout << "Для цикла WHILE рез.    = " << first\_ans << " n = " << n1 << endl;

            } else {

                cout << "Для цикла WHILE точность не достигнута" << endl;

            }

            //Решение циклом do..while

            second\_ans = lim\_doWhile(y0, y1, y2, eps, n2 = 2);

            if (n2 != INT\_MAX) {

                cout.precision(4);  //число знаков после десятичной точки

                cout << "Для цикла DO..WHILE рез.= " << second\_ans << " n = " << n2 << endl;

            } else {

                cout << "Для цикла DO..WHILE точность не достигнута" << endl;

            }

            //Решение циклом for

            third\_ans = lim\_for(y0, y1, y2, eps, n3 = 3);

            if (n3 != INT\_MAX) {

                cout.precision(4);  //число знаков после десятичной точки

                cout << "Для цикла FOR рез.      = " << third\_ans << " n = " << n3 << endl;

            } else {

                cout << "Для цикла FOR точность не достигнута" << endl;

                break;

            }

        default: return 0;

        }//switch

    }//for

}

double lim\_while(double y0, double y1, double y2, double eps, int& n) {

    double y = (y0 + 0.1 \* y2) / (2.5 \* n \* n \* n + 1);

    while (abs(y - y2) > eps && n < INT\_MAX) {

        y0 = y1;

        y1 = y2;

        y2 = y;

        n++;

        y = (y0 + 0.1 \* y2) / (2.5 \* n \* n \* n + 1);

    }

    return y;

}

double lim\_doWhile(double y1, double y2, double y, double eps, int& n) {

    double y0;

    do {

        y0 = y1;

        y1 = y2;

        y2 = y;

        n++;

        y = (y0 + 0.1 \* y2) / (2.5 \* n \* n \* n + 1);

    } while (abs(y - y2) > eps && n < INT\_MAX);

    return y;

}

double lim\_for(double y0, double y1, double y2, double eps, int& n) {

    double y = (y0 + 0.1 \* y2) / (2.5 \* n \* n \* n + 1);

    for (; abs(y - y2) > eps && n < INT\_MAX; ) {

        y0 = y1;

        y1 = y2;

        y2 = y;

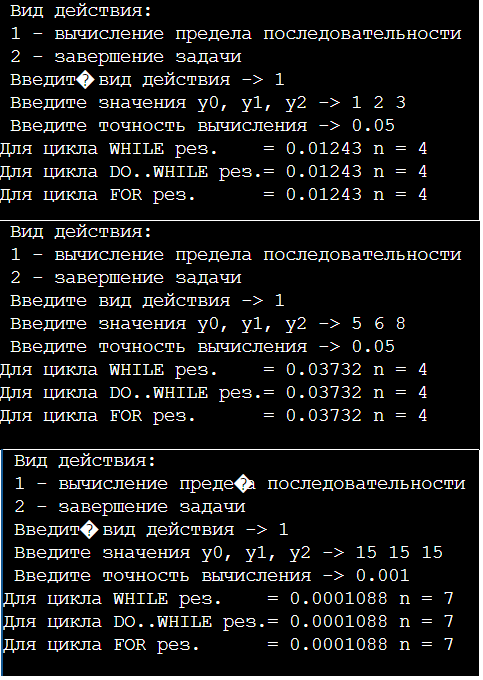
        n++;

        y = (y0 + 0.1 \* y2) / (2.5 \* n \* n \* n + 1);

    }

    return y;

}

***Скриншот результатов выполнения программы***

***Вывод:***

*Я изучил концепции и освоил технологии процедурного программирования, приобрел навыки процедурного программирования на языке C/C++ циклических вычислений.*