МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

* + 1. 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 выполнила

студентка гр. 4141 А. В. Мазанова

Санкт-Петербург

2022

**Вариант 23**

**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).

**Ход выполнения работы:**

1. **Получила индивидуальное задание:**

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

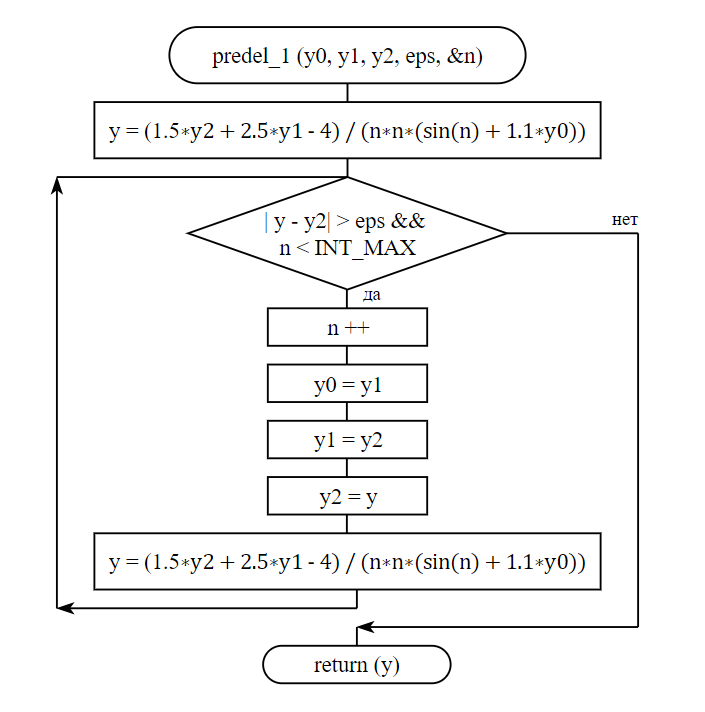


Значения *Y*0, *Y*1, *Y*2 и точность вычисления *ε* вводятся с клавиатуры. Вычисления прекратить при выполнении условия ⎪*Yn*-*Yn*-1⎪≤*ε*.

1. **Разработала математическую модель:**
2. **Математическая модель с использованием цикла while:**

При вызове в функцию *predel\_*1 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 predel\_*1*)***



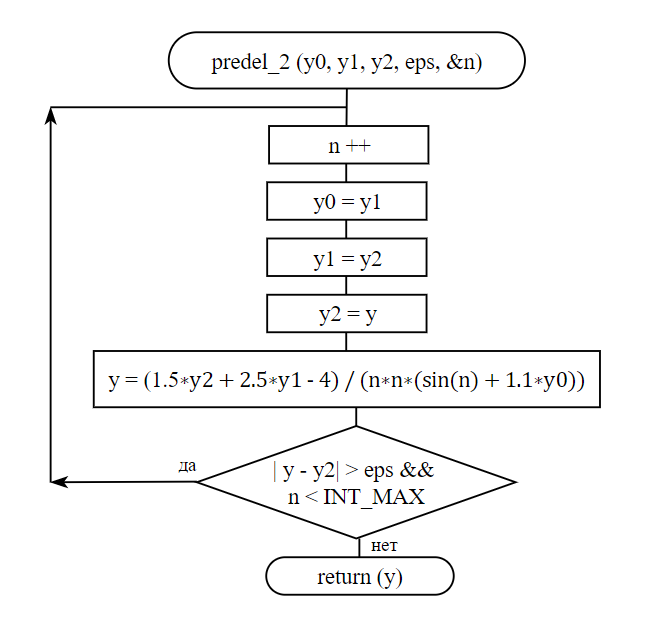
1. ***Математическая модель решения с использованием цикла do…while***

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 Y2, перед этим значение Y2 переписывается в Y1, перед этим значение Y1 переписывается в Y0, 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 predel\_*2*)***

****

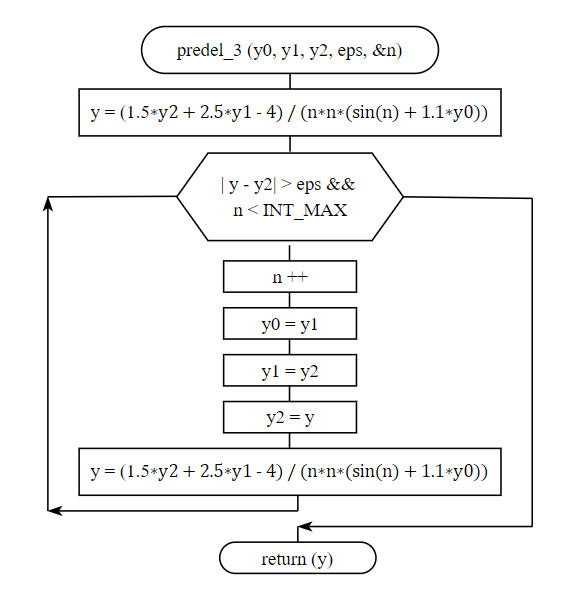
1. ***Математическая модель решения с использованием цикла for***

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 функции for (функция predel\_3)***

******

1. **Составила программу на языке *С/С++***

//Циклические вычисления с заданной точностью. Пример.

//Вычислить предел последовательности {y[n]}.

//Значения y[0], y[1], y[2] и точность eps вводятся с клавиатуры.

//Вычисления закончить при |y[n]-y[n-1]| <= eps.

//Вывести результат и число итераций.

# include <iostream>

# include <cmath>

# include <climits>

using namespace std;

double predel\_1(double, double, double, double, int& n);

double predel\_2(double, double, double, double, int& n);

double predel\_3(double, double, double, double, int& n);

//main\_begin

int main()

{

setlocale(LC\_ALL, "Russian");

int var,

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

double re1, re2, re3, //результаты расчетов

y, //значение предела

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

double eps; //точность вычисления предела

for (;;)

{//Ввод исходных данных

cout << " Вид действия:" << endl;

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

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

cout << " Введите вид действия -> ";

cin >> var;

switch (var)

{

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

re1 = predel\_1(y0, y1, y2, eps, n1 = 3);

if (n1 != INT\_MAX)

{

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

cout << "Для цикла WHILE результат = " << re1 << " n = " << n1 << endl;

}

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

//Решение циклом do..while

re2 = predel\_2(y0, y1, y2, eps, n2 = 2);

if (n2 != INT\_MAX)

{

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

cout << "Для цикла DO..WHILE результат = " << re2 << " n = " << n2 << endl;

}

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

//Решение циклом for

re3 = predel\_3(y0, y1, y2, eps, n3 = 3);

if (n3 != INT\_MAX)

{

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

cout << "Для цикла FOR результат = " << re3 << " n = " << n3 << endl;

}

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

break;

default: return 0;

}//switch

}//for

}

//main\_end

//predel\_1\_beg

//вычисление предела последовательности циклом while

double predel\_1(double y0, double y1, double y2, double eps, int& n)

{

double y = (1.5 \* y2 + 2.5 \* y1 - 4) / (n \* n \* (sin(n) + 1.1 \* y0));

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

{

y0 = y1;

y1 = y2;

y2 = y;

n++;

y = (1.5 \* y2 + 2.5 \* y1 - 4) / (n \* n \* (sin(n) + 1.1 \* y0));

}

return y;

}

//predel\_1\_end

//predel\_2\_beg

//вычисление предела последовательности циклом do..while

double predel\_2(double y1, double y2, double y, double eps, int& n)

{

double y0;

do {

y0 = y1;

y1 = y2;

y2 = y;

n++;

y = (1.5 \* y2 + 2.5 \* y1 - 4) / (n \* n \* (sin(n) + 1.1 \* y0));

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

return y;

}

//predel\_2\_end

//predel\_3\_beg

//вычисление предела последовательности циклом for

double predel\_3(double y0, double y1, double y2, double eps, int& n)

{

double y = (1.5 \* y2 + 2.5 \* y1 - 4) / (n \* n \* (sin(n) + 1.1 \* y0));

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

{

y0 = y1;

y1 = y2;

y2 = y;

n++;

y = (1.5 \* y2 + 2.5 \* y1 - 4) / (n \* n \* (sin(n) + 1.1 \* y0));

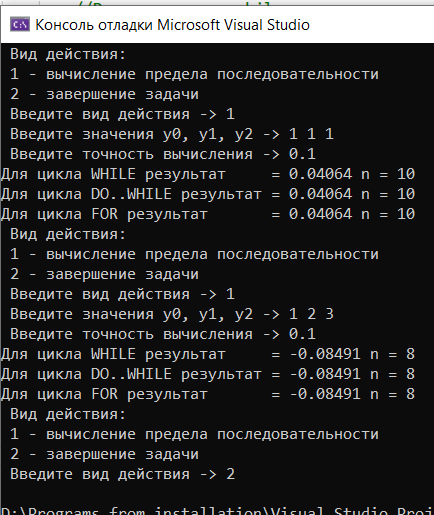
}

return y;

}

//predel\_3\_end

1. **Предоставила результат работы программы на языке *С/С++***

****

1. **Вывод:** я изучила концепции и освоила технологии процедурного программирования, приобрела навыки процедурного программирования на языке *С/С++* циклических вычислений.