|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине: «Основы программирования»

|  |  |  |
| --- | --- | --- |
| Студент |  | Камалов Антон Павлович |
| Группа |  | РК6-15Б |
| Тип задания |  | Лабораторная работа №4 |
| Тема лабораторной работы |  | Циклы |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2022 г.*

# Цель работы

# Изучение управляющих операторов цикла. Знакомство с понятием "точность вычисления".

# Ответы на контрольные вопросы

1. Сначала вычисляется выражение. Если оно истинно, то выполняется оператор. Выражение вновь вычисляется до тех пор, пока оно не станет ложным и произойдёт выход из цикла.
2. Принцип работы как у цикла ***while***, за исключением того, что значение выражения проверяется после выполнения оператора.
3. Цикл for состоит из трёх выражений – первое часто применяется для инициализации переменных, второе используется в качестве условия окончания цикла, а третье для изменения переменной, инициализированной в первом выражении.

Первое выражение вычисляется один раз при входе в цикл; второе используется многократно как условие окончания; третье вычисляется многократно после выполнения операторов тела цикла.

4. int i= 0;

While (i<=10){

I++;

I+=I;

}

Do {

Int j = 0

j++

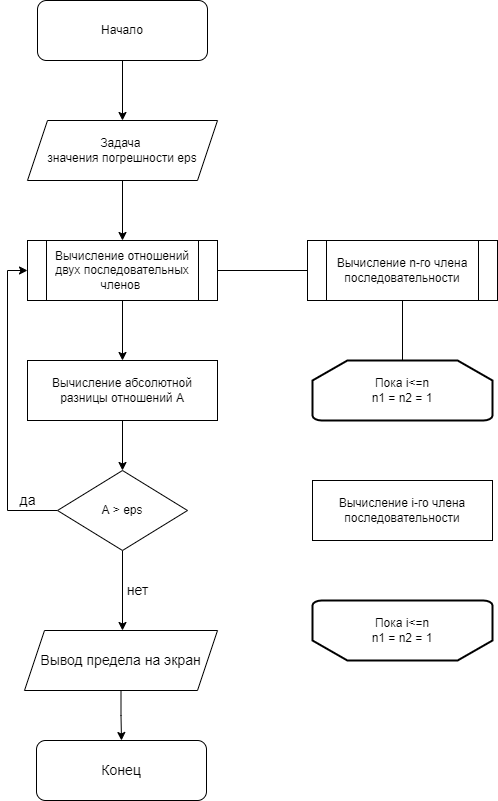
j+=i} while (i<=10)

For (int I = 0; i<=10; i++)

I+=i

1. Точность вычислений – это близость измеренного значения к истинному значению измеряемой величины.

# Блок-схема программы



# Текст программы

# #include <stdio.h>

# #include <math.h>

# #define eps 0.01

# // Ряд Фибоначчи: 1 1 2 3 5 8 13 21

# int fib(int n){ // получение n-го числа Фибоначчи

# int a1 = 1, a2 = 1;

# for (int i = 3; i <= n; i++){

# a2 = a1 + a2;

# a1 = a2 - a1;

# }

# return a2;

# }

# double ot(double n2, double n1){ // получение отношения двух послед. чисел

# double o;

# o = ((double)fib(n2)) / (fib(n1));

# return o;

# }

# void predel(){ // вычисление коэффциента

# int i = 2;

# double f2 = ot(i + 1, i + 2);

# double f1 = ot(i, i+1);

# double A = fabs(f2 - f1);

# while (A > eps){

# i++;

# f2 = ot(i + 1, i + 2);

# f1 = ot(i, i+1);

# A = fabs(f2 - f1);

# }

# printf("%g\n", f2+1); // +1, так как выводится обратный коэффициент

# }

# int main() {

# predel();

# }

# Результаты отладки

После первой компиляции была найдена синтаксическая ошибка: пропущена «;» после вызова функции ***predel***. Также была допущена семантическая ошибка: получение абсолютного значения производилось с помощью функции ***abs***, но результат вывода получался неточным. Исправить её смогла замена ***abs*** на ***fabs***.

# Результаты работы программы

* При eps = 0.01 выводится 1.61538
* При eps = 0.001 выводится 1.61818
* При eps = 0.0001 выводится 1.61806

# Выводы по работе

Я изучил управляющие операторы циклов и познакомился с понятием "точность вычисления".