Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 9

По дисциплине «Основы алгоритмизации и программирования»

На тему «Решение инженерных задач на основе циклических программ»

Выполнила:

Студентка 1 курса 7 группы

Бабич Виолетта Станиславовна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |  |
| --- | --- | --- |
| 2 | cos3(x) | a = 4, b = 7 |

Метод трапеций

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a=4, b=7, h, x1, y1, x2, y2, s = 0;

h = (b - a) / 200; // находим высоту маленьких трапеций

//Находим х1 и у1

x1 = a;

y1 = pow(cos(x1), 3);

//Перебираем все возможные значения х и нахадим сумму трапеций

while (x1 <= b - h) {

x2 = x1 + h;

y2 = pow(cos(x2), 3);

s += h \* (y1 + y2) / 2; // суммируем площади маленьких трапеций

x1 = x2;

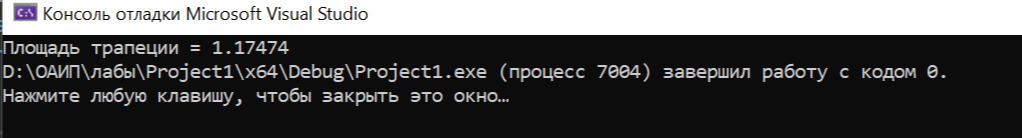
y1 = y2;

}

//Выводим результат

cout << "Площадь трапеции = " << s;

}

****

Метод парабол

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double s = 0, x1, y1, h, a=4, b=7, s1 = 0, s2 = 0, i = 1;

h = (b - a) / (200 \* 2); // находим высоту маленьких трапеций

x1 = a + 2 \* h; //находим х1

//Перебираем числа

while (i < 200) {

y1 = pow(cos(x1), 3);

s2 += y1;

x1 += h;

y1 = pow(cos(x1), 3);

s1 += y1;

x1 += h;

i += 1;

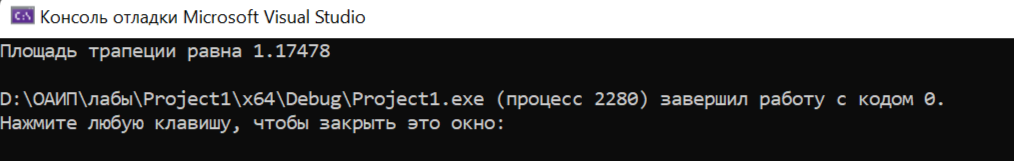
}

//Считаем площадь исрользуя метод парабол

s = (h / 3) \* ((pow(cos(a), 3)) + 4 \* (pow(cos(a + h), 3)) + 4 \* s1 + 2 \* s2 + (pow(cos(b), 3)));

cout << "Площадь трапеции равна " << s << endl;

}



|  |  |
| --- | --- |
| 2 | cos(x) + x – 7 |

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a=4, b=7, e = 0.0001, x, y1, y2;

//Перебираем числа пока модуль из a и b больше чем 2\*е

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

// значение функции от x

y1 = cos(x) + x - 7;

// значение функции от a

y2 = cos(a) + a - 7;

//Сравниваем занчение с нулём, чтобы понять, что является корнем

if (y1 \* y2 <= 0) {

b = x;

}

else {

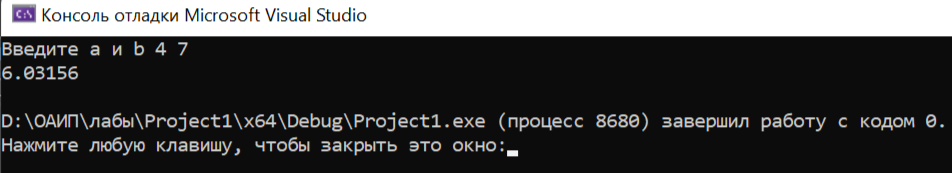
a = x;

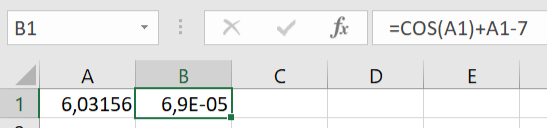
}

}

//Выводим корень

cout << x << endl;

}



Дополнительные задания

|  |  |  |
| --- | --- | --- |
| 1 | x3 – 3 | a = 1, b = 3 |

Метод трапеций

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a, b, n = 200, h, x1, y1, x2, y2, s = 0;

cout << "Введите границы отрезка" << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

h = (b - a) / n; // находим высоту маленьких трапеций

//Считаем х1 и у1

x1 = a;

y1 = pow(x1, 3) - 3;

//Перебираем все возможные х

while (x1 <= b - h) {

x2 = x1 + h;

y2 = pow(x2, 3) - 3;

s += h \* (y1 + y2) / 2; // суммируем площади маленьких трапеций

x1 = x2;

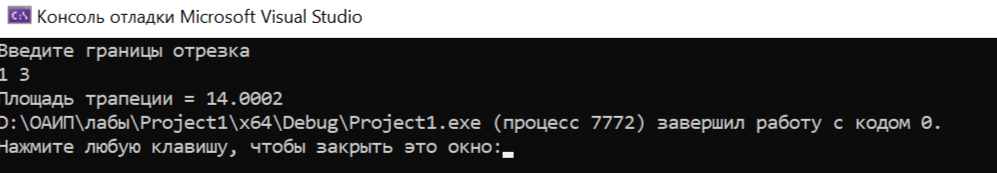
y1 = y2;

}

//Выводим площадь трапеции

cout << "Площадь трапеции = " << s;

}



Метод парабол

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double s = 0, x1, y1, h, a, b, s1 = 0, s2 = 0, i = 1;

cout << "Введите значение a и b " << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

h = (b - a) / (200 \* 2); // находим высоту маленьких трапеций

x1 = a + 2 \* h;

//Перебираем числа,которые будут удовлетворять условию

while (i < 200) {

y1 = pow(x1, 3) - 3;

s2 += y1;

x1 += h;

y1 = pow(x1, 3) - 3;

s1 += y1;

x1 += h;

i += 1;

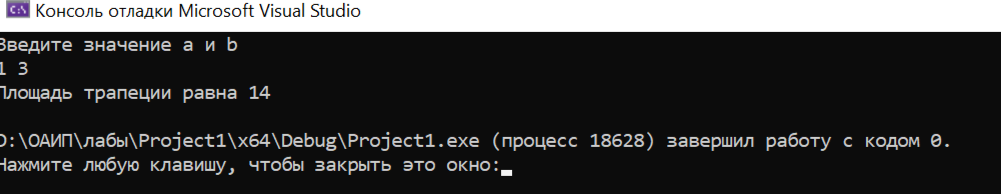
}

//Считаем площадь трапеции, используя метод парабол

s = (h / 3) \* ((pow(a, 3) - 3) + 4 \* (pow(a + h, 3) - 3) + 4 \* s1 + 2 \* s2 + (pow(b, 3) - 3)); // считаем общую площадь

cout << "Площадь трапеции равна " << s << endl;

}



|  |  |  |
| --- | --- | --- |
| 1 | x3 + x – 3 |  |

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001, x, y1, y2;

cout << "Введите значение a и b " << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

//Перебираем числа пока модуль из a и b больше чем 2\*е

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

y1 = pow(x, 3) + x - 3; // значение функции от x

y2 = pow(a, 3) + a - 3; // значение функции от a

//Сравниваем занчение с нулём, чтобы понять, что является корнем

if (y1 \* y2 <= 0) {

b = x;

}

else {

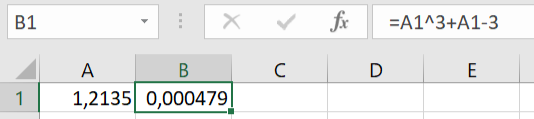
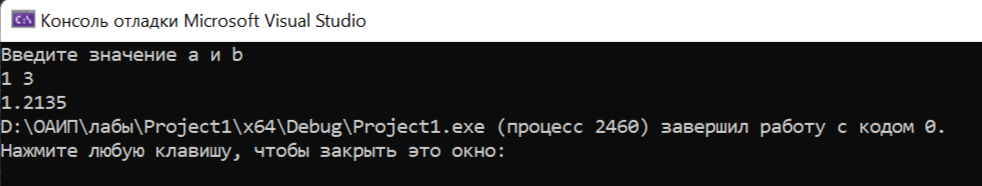
a = x;

}

}

cout << x; // выводим корень

}



|  |  |  |
| --- | --- | --- |
| 3 | 1 +x3 | a = 1, b = 6 |

Метод трапеций

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a, b, n = 200, h, x1, y1, x2, y2, s = 0;

cout << "Введите границы отрезка" << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

h = (b - a) / n; // находим высоту маленьких трапеций

//Считаем х1 и у1

x1 = a;

y1 = 1 + pow(x1, 3);

//Перебираем все возможные х

while (x1 <= b - h) {

x2 = x1 + h;

y2 = 1 + pow(x2, 3);

s += h \* (y1 + y2) / 2; // суммируем площади маленьких трапеций

x1 = x2;

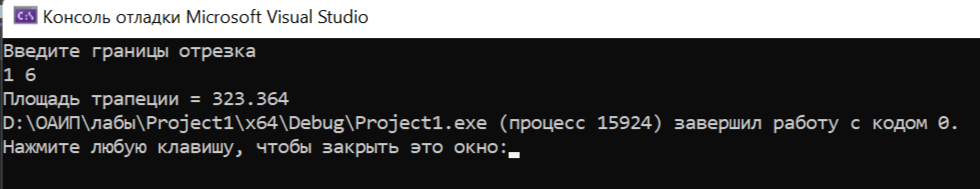
y1 = y2;

}

//Выводим площадь трапеции

cout << "Площадь трапеции = " << s;

}



Метод парабол

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double s = 0, x1, y1, h, a, b, s1 = 0, s2 = 0, i = 1;

cout << "Введите значение a и b " << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

h = (b - a) / (200 \* 2); // находим высоту маленьких трапеций

x1 = a + 2 \* h;

//Перебираем числа,которые будут удовлетворять условию

while (i < 200) {

y1 = 1 + pow(x1, 3);

s2 += y1;

x1 += h;

y1 = 1 + pow(x1, 3);

s1 += y1;

x1 += h;

i += 1;

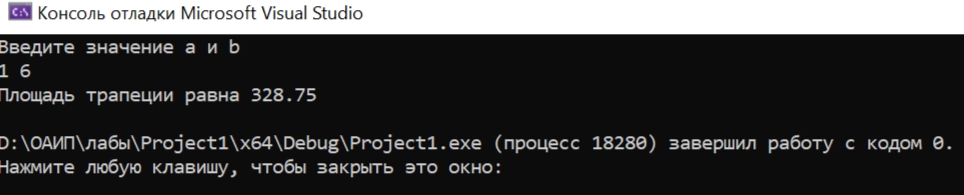
}

//Считаем площадь трапеции, используя метод парабол

s = (h / 3) \* ((1 + pow(a, 3)) + 4 \* (1 + pow(a + h, 3)) + 4 \* s1 + 2 \* s2 + (1 + pow(b, 3))); // считаем общую площадь

cout << "Площадь трапеции равна " << s << endl;

}



|  |  |  |
| --- | --- | --- |
| 3 | x3 + 2x – 1 | 9 |

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001, x, y1, y2;

cout << "Введите значение a и b " << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

y1 = pow(x, 3) + 2 \* x - 1; // значение функции от x

y2 = pow(a, 3) + 2 \* a - 1; // значение функции от a

//Сравниваем занчение с нулём, чтобы понять, что является корнем

if (y1 \* y2 <= 0) {

b = x;

}

else {

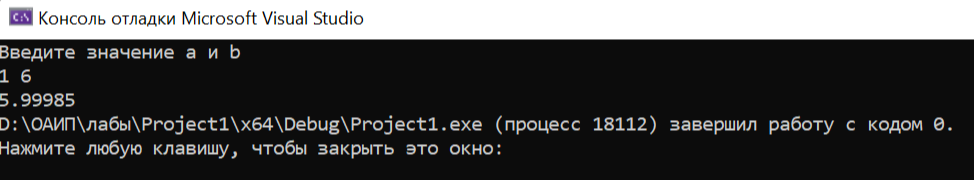
a = x;

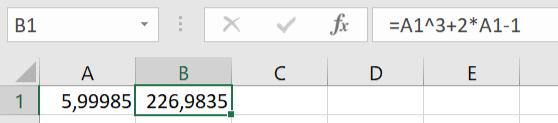
}

}

cout << x; // выводим корень

}





|  |  |  |
| --- | --- | --- |
| 13 | 2+x3 | a=8, b=14 |

Метод трапеций

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a, b, n = 200, h, x1, y1, x2, y2, s = 0;

cout << "Введите границы отрезка" << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

h = (b - a) / n; // находим высоту маленьких трапеций

//Считаем х1 и у1

x1 = a;

y1 = pow(x1, 3) + 2;

//Перебираем все возможные х

while (x1 <= b - h) {

x2 = x1 + h;

y2 = pow(x2, 3) + 2;

s += h \* (y1 + y2) / 2; // суммируем площади маленьких трапеций

x1 = x2;

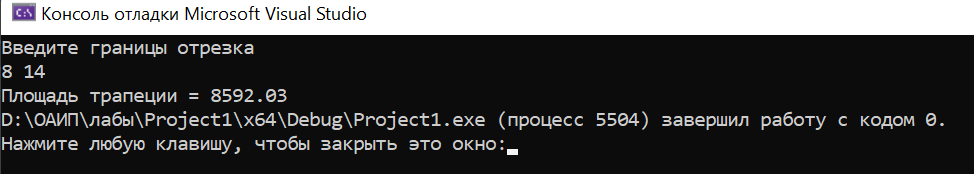
y1 = y2;

}

//Выводим площадь трапеции

cout << "Площадь трапеции = " << s;

}



Метод парабол

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double s = 0, x1, y1, h, a, b, s1 = 0, s2 = 0, i = 1;

cout << "Введите значение a и b " << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

h = (b - a) / (200 \* 2); // находим высоту маленьких трапеций

x1 = a + 2 \* h;

//Перебираем числа,которые будут удовлетворять условию

while (i < 200) {

y1 = 2 + pow(x1, 3);

s2 += y1;

x1 += h;

y1 = 2 + pow(x1, 3);

s1 += y1;

x1 += h;

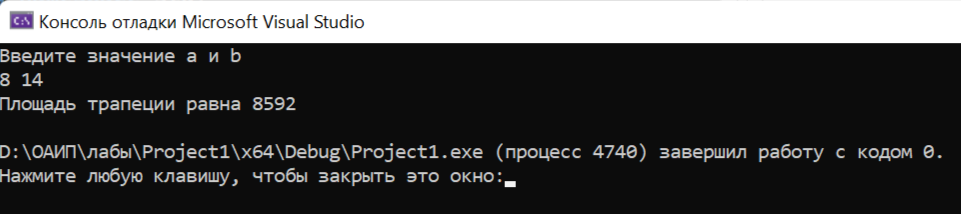
i += 1;

}

//Считаем площадь трапеции, используя метод парабол

s = (h / 3) \* ((2 + pow(a, 3)) + 4 \* (2 + pow(a + h, 3)) + 4 \* s1 + 2 \* s2 + (2 + pow(b, 3))); // считаем общую площадь

cout << "Площадь трапеции равна " << s << endl;

}

|  |  |
| --- | --- |
| 13 | x-4+ex |

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001, x, y1, y2;

cout << "Введите значение a и b " << endl;

cin >> a >> b; // считываем координаты концов отрезка на оси x

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

y1 = pow(e, x) + x - 4; // значение функции от x

y2 = pow(e, a) + a - 4; // значение функции от a

//Сравниваем занчение с нулём, чтобы понять, что является корнем

if (y1 \* y2 <= 0) {

b = x;

}

else {

a = x;

}

}

cout << x; // выводим корень

}

