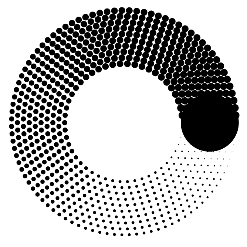
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 8**

**Дисциплина:** Численные методы в компьютерных вычислениях

**Выполнил:** студент группы 231-338

Шаура Илья Максимович

**** (Фамилия И.О.)

**Дата, подпись** 29.03.2024  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Дата) (Подпись)

**Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Фамилия И.О., степень, звание) (Оценка)

**Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Дата) (Подпись)

Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва**

**2024**

**Вариант 14**

Листинг 1 – Код консольной программы

|  |
| --- |
| #include <iostream>  double exact\_value = 15.5687;  double bottom\_value = 1;  double top\_value = 100;  double f(double x) {  return (sqrt(1 + 5 \* log(x))) / x;  }  double middle\_rect(int n) {  double h = (top\_value - bottom\_value) / n;  double it = bottom\_value;  double sum = 0;  for (size\_t i = 0; i < n; i++)  {  sum += f(it + h/2);  it += h;  }  sum \*= h;    return sum;  }  double trapeze(int n) {  double h = (top\_value - bottom\_value) / n;  double it = bottom\_value;  double sum = 0;  for (size\_t i = 0; i < n; i++)  {  sum += (f(it) + f(it+h)) / 2;  it += h;  }    sum \*= h;  return sum;  }  double simpson(int n) {  double h = (top\_value - bottom\_value) / n;  double it = bottom\_value;  double sum = 0;  sum += f(bottom\_value) + f(top\_value);  for (size\_t i = 1; i < n; i++)  {  it += h;  if (i % 2 == 0) sum += 2 \* f(it);  else sum += 4 \* f(it);  }    sum \*= h / 3;  return sum;  }  double average\_errors(float errors[])  {  double sum = 0;  for (int i = 0; i < 9; i++)  sum += abs(errors[i] - errors[i + 1]);  return std::pow(sum / 10, -1);  }  int main()  {  int n[10] = { 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 };  std::cout << "Exact value = " << exact\_value << std::endl;  std::cout << std::endl;  //std::cout << "----- MIDDLE RECTANGLES METHOD -----" << std::endl;  //  //double mr = middle\_rect();  //std::cout << "Solved integral: " << mr << std::endl;  //std::cout << "Error: " << abs(exact\_value - mr) << std::endl;  //std::cout << std::endl;  //std::cout << std::endl;  std::cout << "----- TRAPEZE METHOD -----" << std::endl;  std::cout << "n\tIntegral\tError\t\tAccuracy\n\n";  float tr\_errors[10];  float prev\_error = 0;  for (size\_t i = 0; i < 10; i++)  {  float integral = round(trapeze(n[i]) \* 10000)/10000;  float error = round(abs(exact\_value - integral) \* 10000)/10000;  std::cout << n[i] << "\t" << integral << "\t\t" << error << "\t\t" << prev\_error / error << std::endl;  tr\_errors[i] = error;  prev\_error = error;  }  std::cout << std::endl;  std::cout << std::endl;  std::cout << "----- SIMPSON METHOD -----" << std::endl;    std::cout << "n\tIntegral\tError\t\tAccuracy\n\n";  float sm\_errors[10];  prev\_error = 0;  for (size\_t i = 0; i < 10; i++)  {  float integral = simpson(n[i]);  float error = abs(exact\_value - integral);  std::cout << n[i] << "\t" << integral << "\t\t" << error << "\t\t" << prev\_error / error << std::endl;  sm\_errors[i] = error;  prev\_error = error;  }  std::cout << std::endl;  std::cout << std::endl;  } |