**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Сибирский государственный университет науки и технологий**

**имени академика М.Ф. Решетнёва»**

ИИТК/09.03.01/\Информатика и вычислительная техника/Автоматизированные системы обработки информации и управления

институт/факультет/подразделение

Информатики и вычислительной техники

кафедра

**Определенные интегралы**

Преподаватель \_\_30.03.2022\_\_\_\_\_ Семенкина М.Е.

подпись, дата инициалы, фамилия

Обучающиеся \_\_БИА21-01 \_ \_\_\_30.03.2022\_\_\_

номер группы подпись, дата Е.В.Карапузин,

Р. В. Сухих,

А.С.Мкртчян,

П.А. Хурасев.

инициалы, фамилии

Красноярск 2022

Введение

В нашей работе рассматривается стандартная и хорошо всем знакомая задача – вычисление значений определенных интегралов.

На данный момент не существует универсального метода определения интегралов. Тогда почему бы не использовать сразу несколько подходов?

Нами предполагается возможность вычисления интеграла по желаемой формуле или нескольким.

Для вычислений необходимы лишь некоторые данный: границы интегрирования и точность.

Цели

* Изучить различные способы вычисления определенного интеграла.
* Выделить три наиболее актуальных способа вычислений.
* Создать программу, использующую эти способы, которая поможет пользователю получить значение определенного интеграла определенным способом.

Ход работы

Функция, отвечающая за открытие интерфейса программы:

[***#pragma***](https://vk.com/im?sel=179355208&st=%23pragma)***endregion  
private: System::Void pictureBox1\_Click(System::Object^ sender, System::EventArgs^ e) {  
Close();  
}***

Функция, отвечающая за сворачивание программы:

***private: System::Void pictureBox2\_Click(System::Object^ sender, System::EventArgs^ e) {  
this->WindowState = FormWindowState::Minimized;  
}***

Функция, отвечающая за перемещение интерфейса программы.С ее помощью мы имеем возможность корректировать местоположение окна программы во время работы с ней:

***private: System::Void Panel\_MouseDown(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {  
this->dragging = true;  
this->offset = Point(e->X, e->Y);  
}  
private: bool dragging;  
private: Point offset;  
private: System::Void Shell\_Load(System::Object^ sender, System::EventArgs^ e) {  
this->dragging = false;  
}  
private: System::Void Panel\_MouseMove(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {  
if (this->dragging) {  
Point currentScreenPos = PointToScreen(e->Location);  
Location = Point(currentScreenPos.X - this->offset.X, currentScreenPos.Y - this->offset.Y);  
}  
};  
  
private: System::Void Panel\_MouseUp(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {  
this->dragging = false;  
}***

Функция блокировки работы программы при наличии пустых (незаполненных) полей.То есть, при наличии незаполненных данных программа работать не будет.

***private: System::Void panel3\_Click(System::Object^ sender, System::EventArgs^ e) {  
bool f = true;***

Очищение точек на графике представлено следующими строками системного кода:

***chart1->Series[0]->Points->Clear();  
chart1->Series[1]->Points->Clear();  
chart1->Series[2]->Points->Clear();***

Объявление конвертации строк:

***marshal\_context^ marshal = gcnew marshal\_context();  
if (textBox3->Text != "") accuracy = atof(marshal-***

Считывание поля «точность» из текстбокса, задается в переменную:

***>marshal\_as<const char\*>(textBox3->Text));  
else f = false;  
if (textBox1->Text != "") value1 = atof(marshal->marshal\_as<const char\*>(textBox1->Text));  
else f = false;  
if (textBox2->Text != "") value2 = atof(marshal->marshal\_as<const char\*>(textBox2->Text));  
else f = false;  
double n = (value2 - value1) / accuracy; // задаём число разбиений n***

Далее программа занимается поиском интеграла в соответствии с заданным способом.

***int k = 0; //счётчик точек для графика  
if (f) {  
for (; value1 <= value2; value1 += accuracy) {  
for (int i = 1; i <= n; i++) { //интегралл первым способом  
Integral1 += accuracy \* pow(value1 + accuracy \* (i - 0.5), 2);  
}  
  
Integral2 = accuracy \* (pow(value1, 2) + pow(value2, 2)) / 2.0;  
for (int i = 1; i <= n - 1; i++) { //интегралл вторым способом  
Integral2 += accuracy \* pow(value1 + accuracy \* i, 2);  
}  
  
Integral3 = accuracy \* (pow(value1, 2) + pow(value2, 2)) / 6.0; //интегралл третьим способом  
for (int i = 1; i <= n; i++) {  
Integral3 += 4.0 / 6.0 \* accuracy \* pow(value1 + accuracy \* (i - 0.5), 2);  
}***

***for (int i = 1; i <= n - 1; i++) {  
Integral3 += 2.0 / 6.0 \* accuracy \* pow(value1 + accuracy \* i, 2);  
}***

Добавление новой точки на графике в соответствии со способом задания. Integral1 – ось X, Value1 – ось Y.Далее интегралы обнуляются для поиска следующей точки.

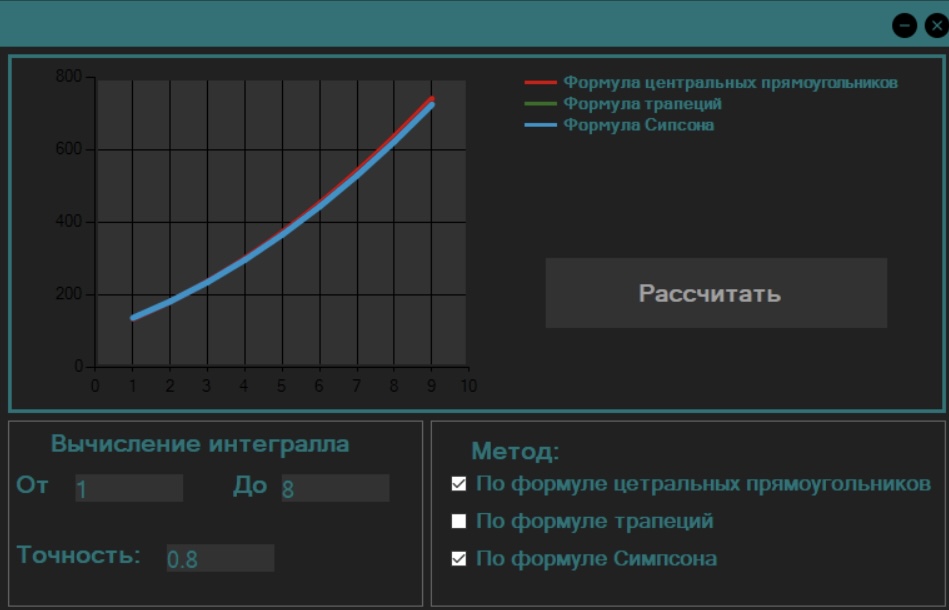
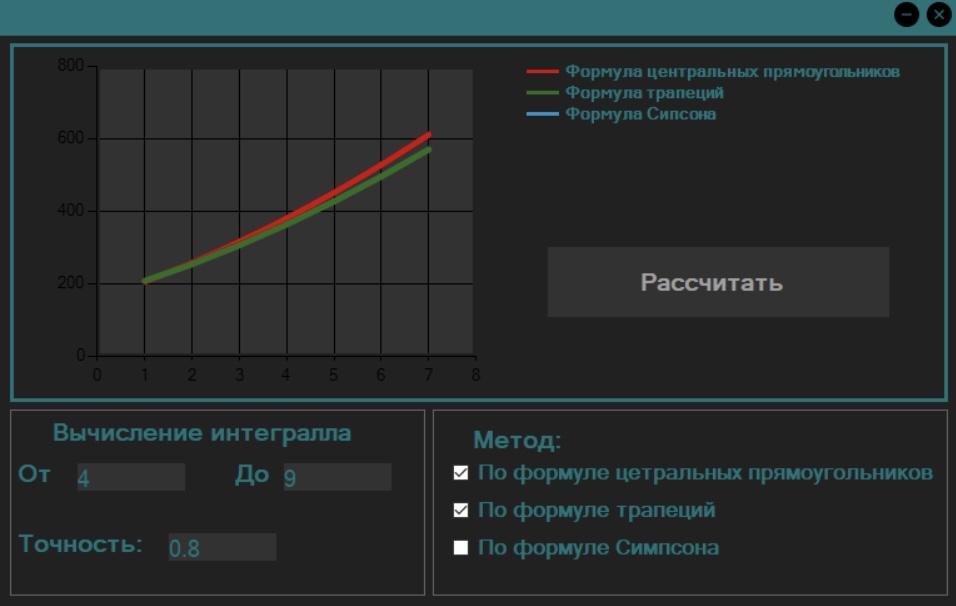
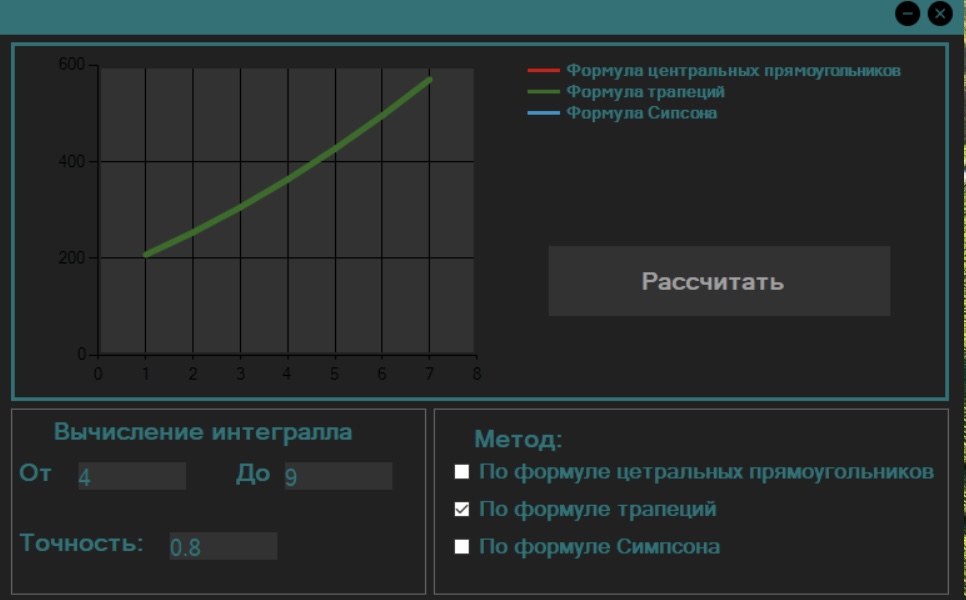
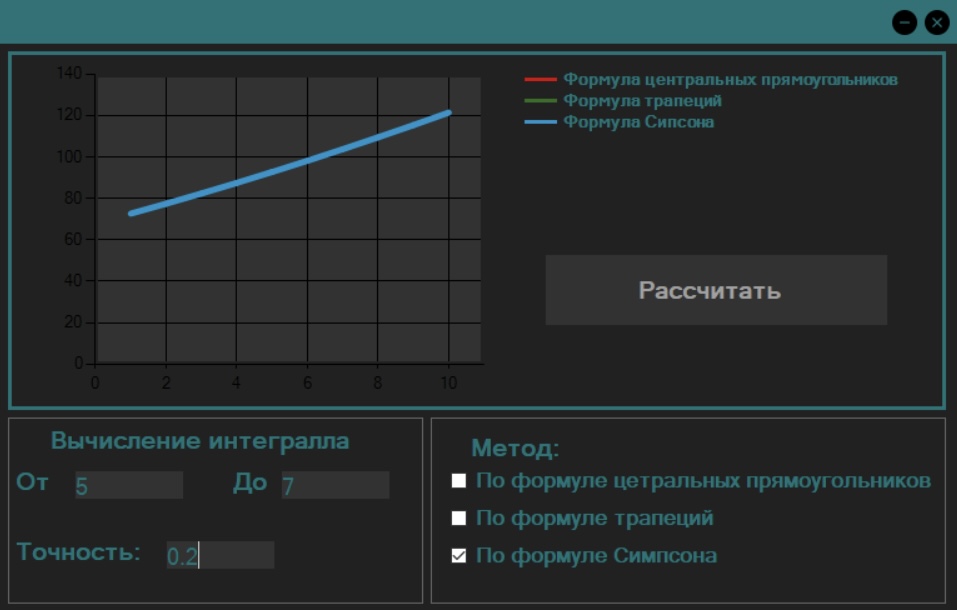
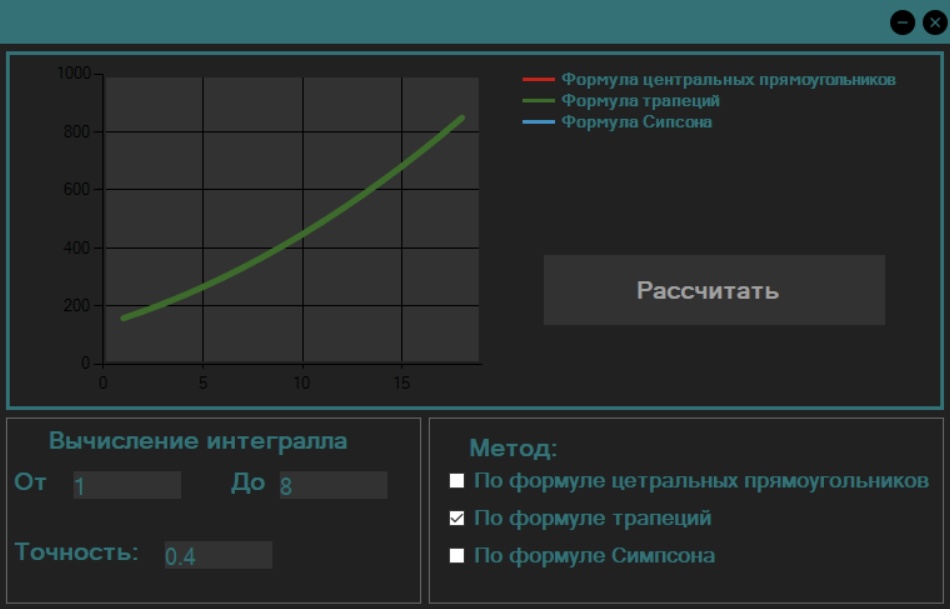
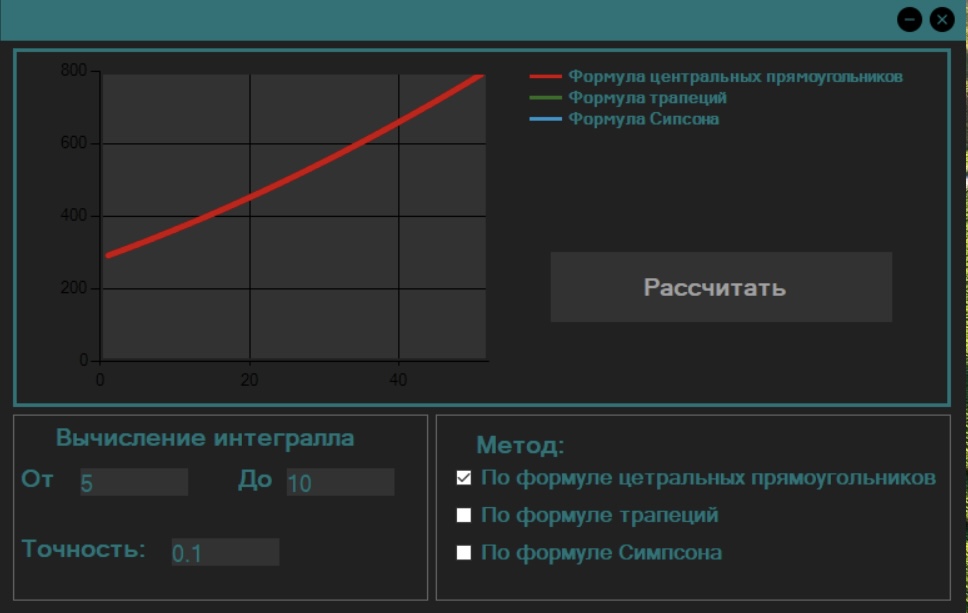
***if (checkBox1->Checked) chart1->Series[0]->Points->Add(Integral1, value1);  
if (checkBox2->Checked) chart1->Series[1]->Points->Add(Integral2, value1);  
if (checkBox3->Checked) chart1->Series[2]->Points->Add(Integral3, value1);  
k++;  
Integral1 = 0.;  
Integral2 = 0.;  
Integral3 = 0.;******}  
  
}  
}  
};  
}***

ВАЖНО! Чем больше точность, тем нагляднее видно, что графики не соприкасаются друг с другом.В обратном же случае графики фактически лежат друг на друге.

Код программы

[#pragma](https://vk.com/im?sel=179355208&st=%23pragma) endregion  
private: System::Void pictureBox1\_Click(System::Object^ sender, System::EventArgs^ e) {  
Close();  
}  
private: System::Void pictureBox2\_Click(System::Object^ sender, System::EventArgs^ e) {  
this->WindowState = FormWindowState::Minimized;  
}  
  
private: System::Void Panel\_MouseDown(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {  
this->dragging = true;  
this->offset = Point(e->X, e->Y);  
}  
private: bool dragging;  
private: Point offset;  
private: System::Void Shell\_Load(System::Object^ sender, System::EventArgs^ e) {  
this->dragging = false;  
}  
private: System::Void Panel\_MouseMove(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {  
if (this->dragging) {  
Point currentScreenPos = PointToScreen(e->Location);  
Location = Point(currentScreenPos.X - this->offset.X, currentScreenPos.Y - this->offset.Y);  
}  
};  
  
private: System::Void Panel\_MouseUp(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e) {  
this->dragging = false;  
}  
  
private: System::Void panel3\_Click(System::Object^ sender, System::EventArgs^ e) {  
bool f = true;  
chart1->Series[0]->Points->Clear();  
chart1->Series[1]->Points->Clear();  
chart1->Series[2]->Points->Clear();  
  
marshal\_context^ marshal = gcnew marshal\_context();  
if (textBox3->Text != "") accuracy = atof(marshal->marshal\_as<const char\*>(textBox3->Text));  
else f = false;  
if (textBox1->Text != "") value1 = atof(marshal->marshal\_as<const char\*>(textBox1->Text));  
else f = false;  
if (textBox2->Text != "") value2 = atof(marshal->marshal\_as<const char\*>(textBox2->Text));  
else f = false;  
double n = (value2 - value1) / accuracy; // задаём число разбиений n  
  
int k = 0; //счётчик точек для графика  
if (f) {  
for (; value1 <= value2; value1 += accuracy) {  
for (int i = 1; i <= n; i++) { //интегралл первым способом  
Integral1 += accuracy \* pow(value1 + accuracy \* (i - 0.5), 2);  
}  
  
Integral2 = accuracy \* (pow(value1, 2) + pow(value2, 2)) / 2.0;  
for (int i = 1; i <= n - 1; i++) { //интегралл вторым способом  
Integral2 += accuracy \* pow(value1 + accuracy \* i, 2);  
}  
  
Integral3 = accuracy \* (pow(value1, 2) + pow(value2, 2)) / 6.0; //интегралл третьим способом  
for (int i = 1; i <= n; i++) {  
Integral3 += 4.0 / 6.0 \* accuracy \* pow(value1 + accuracy \* (i - 0.5), 2);  
}  
  
for (int i = 1; i <= n - 1; i++) {  
Integral3 += 2.0 / 6.0 \* accuracy \* pow(value1 + accuracy \* i, 2);  
}  
  
if (checkBox1->Checked) chart1->Series[0]->Points->Add(Integral1, value1);  
if (checkBox2->Checked) chart1->Series[1]->Points->Add(Integral2, value1);  
if (checkBox3->Checked) chart1->Series[2]->Points->Add(Integral3, value1);  
k++;  
Integral1 = 0.;  
Integral2 = 0.;  
Integral3 = 0.;  
}  
  
}  
}  
};  
}

Работа программы



*Интерфейс и работы программы (Скриншоты 1-6)*

Заключение

Итогом нашей работы стала простая для понимания программа, позволяющая пользователю получить искомое значение интеграла любым из доступных способов. Также представляется наглядный график интеграла. Это позволит существенно сократить время вычислений и позволяет подобрать наиболее подходящий способ вычисления.