**Лабораторная работа №9**

**Задание:**

Создайте графическое приложение, вычисляющее площадь круга методом Монте-Карло. Для этого необходимо создать две фигуры: квадрат и круг, вписанный в данный квадрат, и в случайном порядке размещать внутри квадрата точки.

Площадь круга будет вычисляться по формуле: площадь квадрата \* количество точек внутри круга / общее количество точек внутри квадрата.

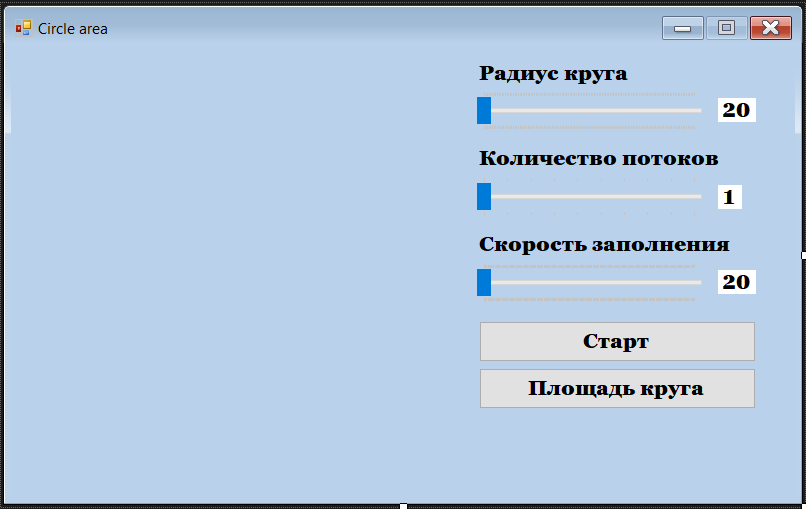
Каждый поток генерирует очередную точку, отрисовывает её и засыпает на установленный пользователем промежуток времени. Для разных потоков цвета точек должны отличаться.

Пользовательский интерфейс должен поддерживать возможность:

* установки радиуса круга;
* установки количества потоков;
* установки скорости заполнения квадрата точками (т.е. промежутка времени, на который засыпает каждый поток);
* отображения квадрата, круга и сгенерированных точек;
* отображения результата (площади круга).

**Ход работы**:

Создадим форму для данной задачи. Скроллбоксами выбираем радиус круга, количество потоков, скорость заполнения.



При нажатии на кнопку Старт выключаются все скроллбоксы и сама кнопка, рисуются круг соответствующего радиуса и описанный квадрат. Для рассчёта в потоках попадания произвольной точки в круг сохраняем координаты центра окружности. Рассчитываем общее количество точек, которое будет выведено потоками – это количество потоков, умноженное на количество точек, выводимое одним потоком (200). Запускаем цикл, который создает заданное количество потоков и передаем им на выполнение метод Count. Задержка нужна для того, чтобы создались все потоки (без задержки потоки иногда не создаются).

private void buttonStart\_Click(object sender, EventArgs e)

{

buttonStart.Enabled = false;

trackBarRadius.Enabled = false;

trackBarCnt.Enabled = false;

trackBarSpeed.Enabled = false;

CreateGraphics().DrawRectangle(new Pen(Color.Black, 1), 1, 20, 2 \* radius, 2 \* radius);

CreateGraphics().DrawEllipse(new Pen(Color.Black, 1), 1, 20, 2 \* radius, 2 \* radius);

x\_c = 1 + radius;

y\_c = 20 + radius;

cnt\_all = 200\*cnt\_streams;

for (ushort i = 1; i <= cnt\_streams; i++)

{

ThreadPool.QueueUserWorkItem(Count);

Thread.Sleep(20);

}

//Thread.Sleep((230 - speed)\*150);

//labelArea.Text = (radius \* radius \* 4 \* cnt / cnt\_all).ToString();

}

Рассмотрим метод Count. Каждому потоку задается произвольный цвет, и через цикл мы создаем 200 точек с произвольными координатами. Если точка попадает в круг, то мы увеличиваем соответствующий счётчик. Скорость заполнения регулируется параметром, задаваемым пользователем. Из 220 мы вычитаем этот параметр, тем самым задавая задержку между выводом точек.

private void Count(object o)

{

Random rnd = new Random();

int R = rnd.Next(0, 255);

int G = rnd.Next(0, 255);

int B = rnd.Next(0, 255);

for (int i = 1; i <= 200; i++)

{

int x = rnd.Next(1, 1 + 2 \* radius);

int y = rnd.Next(20, 20 + 2 \* radius);

if ((x - x\_c) \* (x - x\_c) + (y - y\_c) \* (y - y\_c) <= radius \* radius) cnt++;

CreateGraphics().DrawEllipse(new Pen(Color.FromArgb(R, G, B), 2), x, y, 2, 2);

Thread.Sleep(220 - speed);

}

}

После расстановки всех точек нажимаем на кнопку «Площадь круга», и нам выскакивает сообщение с полученной площадью.

private void buttonArea\_Click(object sender, EventArgs e)

{

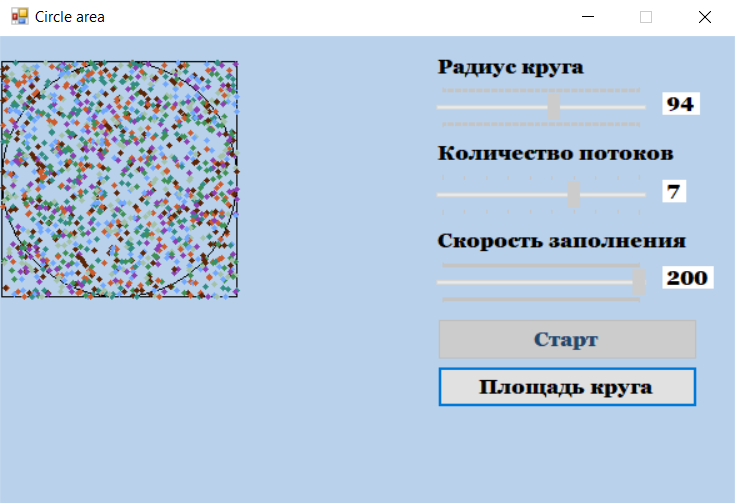
area = radius \* radius \* 4 \* cnt / cnt\_all;

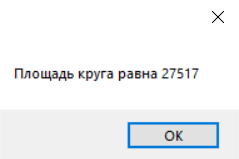
MessageBox.Show("Площадь круга равна " + area);

}

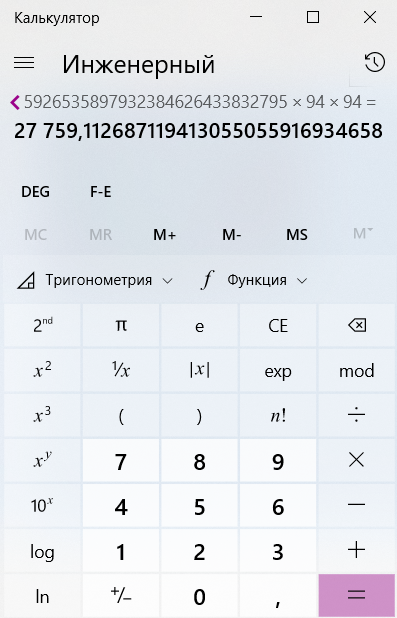
**Демонстрация**:

Рассчитаем площадь круга с радиусом 94, используя 7 потоков и максимальную скорость заполнения.





Проверим посчитанное программой значение с помощью формулы площади круга .



Получили значение 27 759, что подтверждает корректность работы программы (все потоки работают).