|  |
| --- |
| **Лабораторная работа № 11** |
| **Тема** Использование языка программирования высокого уровня для создания Windows-приложения вычисляющего площадь криволинейной фигуры по методу Монте-Карло |
| **Цель выполнения работы**: поэтапное формирование компетенций ОК.2, ОК.4,ПК.3.1 |
| **Оборудование** (дидактическое обеспечение – электронное пособие, справочники с формулами): |
|  |
| **Компьютерная программа** Среда программирования Visual StudioCommunity , Тестовый редактор, Электронные таблицы, Редактор диаграмм, калькулятор |

**Теория**

<http://e.lanbook.com/book/94749>

Статистический метод Монте-Карло

История метода

* Создан в 1949 году
* Авторы:
  + Станислав Улам
  + Джон фон Нейман
  + Николас Метрополис
* Теоретические основы метода были известны давно, но до появления ЭВМ были неэффективны. *Название метода происходит от названия города в княжестве Монако, широко известного своими многочисленными казино, поскольку именно рулетка является одним из самых широко известных* ***генераторов случайных чисел.***
* Только после появления ЭВМ метод Монте-Карло стал универсальным числовым методом

Идея метода

|  |  |
| --- | --- |
| 1. Пусть необходимо вычислить площадь криволинейной фигуры –S | S |
| 1. Возьмем объект с легко вычисляемой площадью S0, например прямоугольник или квадрат. *S0=(b-a)(d-c)*   Внутри этой фигуры разместим криволинейную фигуру, площадь которой надо вычислить | S  *c*  *a*  *d*  *b* |

|  |  |
| --- | --- |
| 1. В этот прямоугольник начнем «набрасывать» случайным образом **N** точек со случайными координатами X и Y:   X=a+(b-a)\*rnd  Y=c+(d-c)\*rnd | S  *c*  *a*  *d*  *b* |
| 1. Среди случайных точек, есть те, которые попадают в фигуру. Задав условие принадлежности точки данной фигуре, с помощью счетчика можно сосчитать количество «попавших» | K(счетчик)=0  Для i=1,N  Если {(x,y) ∈ область фигуры}  K=K+1 |
| 1. При большом количестве точек, и хорошем генераторе случайных чисел, точки равномерно распределяются по площади прямоугольника  |  |  | | --- | --- | | Относительная частота попадания в фигуру или вероятность попадания:  P=K/N | В то же время вероятность попадания зависит от площади фигуры:  P=S/S0, |  1. откуда: S= S0\*K/N | |

**Постановка задачи или ситуации :**

Имеется чертеж криволинейной фигуры

На основе приложения « Создание криволинейной фигуры», создать Приложение "Метод Монте-Карло" для определения площади криволинейной фигуры, которое

1. ограничивает созданную фигуру прямоугольником
2. внутри прямоугольника генерирует случайные N точек
3. проверяет принадлежность каждой точки созданной фигуре
4. рисует попавшую точку в фигуру
5. одновременно подсчитывает количество «попавших» точек
6. вычисляет площадь по формуле 
7. выводит площадь в текстовое поле
8. Сохраняет фигуру с заливкой в папку Изображения
9. Оформить отчет по стандарту с рамками в следующей последовательности:
10. Задание
11. Текст программы
12. Образы экрана до и после запуска приложения
13. Скриншот папки Изображения
14. Анализ результатов выполнения программы

**Контрольные вопросы:**

1. Что осуществляет фрагмент программы

for (k = 1; k <= n; k += 1)

a = -99 + rnd.Next(0, 198);

1. Что такое graph1.DrawRectangle(Z, X1, Y1, width, height)?
2. Перечислить аргументы методаDrawArc

Криволинейные фигуры (1клетка=50 пикселей)

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
|  |  |  |
| 4 | 5 | 6 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 7 | 8 | 9 |
|  |  |  |
| 10 | 11 | 12 |
|  |  |  |
| 13 | 14 | 15 |
|  |  |  |
| 16 | 17 | 18 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 19 | 20 | 21 |
|  |  |  |
| 22 | 23 | 24 |
|  |  |  |
| 25 | 26 | 27 |
|  |  |  |

Образец выполнения лабораторной работы для варианта 0:

Тема: Создание приложения "Метод Монте-Карло" для определения площади криволинейной фигуры

Цель: : поэтапное формирование компетенций ОК.2, ОК.4,ПК.3.1.

|  |  |
| --- | --- |
| **. Задание**:На основе приложения «Создание криволинейной фигуры», создать Приложение "Метод Монте-Карло" для определения площади криволинейной фигуры, которое   1. ограничивает созданную фигуру прямоугольником 2. внутри прямоугольника генерирует случайные N точек 3. проверяет принадлежность каждой точки созданной фигуре 4. рисует попавшую точку в фигуру 5. одновременно подсчитывает количество «попавших» точек 6. вычисляет площадь по формуле 7. выводит площадь в текстовое поле 8. Сохраняет фигуру с заливкой в папку Изображения |  |

Расчетные формулы для области:

= (y≤200) and ( X≤0) and or (x2+(y-200)2 ≥ 40000) and (x≥0) and ((x+200)2+(y+200)2≥ 40000) and (x ≥-200)

Контрольный тест

Предварительно вычислить площадь фигуры

Фигуру можно мысленно разбить на 3 части. Площадь двух одинаковых частей получена вычитанием из площади квадрата со стороной 200 четверти круга радиуса 200, Площадь третьей части равна половине площади квадрата со стороной 200:

|  |  |
| --- | --- |
| 200  200 | кв.ед |

Рабочий вид формы

|  |  |  |
| --- | --- | --- |
|  | | Прим. Копировать всю папку «Графические методы» в ту же папку.  Копию переименовать – Метод Монте-Карло.  Далее и форму переименовать в Метод Монте-Карло  На форме разместить Элементы управления textBox1и textBox2 (размер шрифта- 12пт), label1, label2 (12пт) |
| Главное меню дополнить в соответствии с таблицей:   |  |  |  | | --- | --- | --- | | toolStripMenuItem1 | toolStripMenuItem6 |  | | Файл | Фигура |  | | toolStripMenuItem2 | toolStripMenuItem7 |  | | Система координат | Дуги и секторы |  | | toolStripMenuItem3 | toolStripMenuItem8 |  | | Открыть | Путь- Контур |  | | toolStripMenuItem4 | toolStripMenuItem9 |  | | Сохранить | Просто Контур |  | | toolStripMenuItem5 | toolStripMenuItem13 |  | | Выход | HQ Контур |  | |  | toolStripMenuItem16 |  | |  | Сплошная заливка |  | |  | toolStripMenuItem17 |  | |  | Уорная аливка |  | |  | toolStripMenuItem18 |  | |  | Градиентная заливка |  | |  | toolStripMenuItem12 |  | |  | Прямоугольник |  | | Свойство Visible для textBox2 и label2- false | |
| |  |  |  | | --- | --- | --- | | toolStripMenuItem1 | toolStripMenuItem6 | toolStripMenuItem10 | | Файл | Фигура | Вычисление площади | | toolStripMenuItem2 | toolStripMenuItem7 | toolStripMenuItem11 | | Система координат | Дуги и секторы | Площадь | | toolStripMenuItem3 | toolStripMenuItem8 |  | | Открыть | Контур (как путь) |  | | toolStripMenuItem4 | toolStripMenuItem9 |  | | Сохранить | Контур (не путь) |  | | toolStripMenuItem5 | toolStripMenuItem12 |  | | Выход | Прямоугольник |  | |

Программный код : *В разделах: Дуги и секторы, Контуры (двух видов) добавить строки:*

|  |  |
| --- | --- |
| Graph1.PageUnit = GraphicsUnit.Pixel | PageUnit- определяет единицу изменения, которая будет использоваться для выражения координат в объекте Graph1 |
| Graph1.SmoothingMode = SmoothingMode.AntiAlias | SmoothingMode *-Качество визуализации графического объекта* Graph1  *.Default – нет сглаживания*  .AntiAlias- рисование со сглаживанием |
| Или .HighQuality; | .HighQuality-рисование со сглаживанием |

*В результате все кривые и прямые избавятся от растрирования*

1. ***Прямоугольник*** , в который полностью помещается заданная фигура

*Прорисовка прямоугольника, ограничивающего фигуру, где( -200,200) координаты нижнего левого угла, а 200 и 400 ширина и длина прямоугольника*



**Площадь**

*Координаты x,y определяются исходя из координат и размеров прямоугольника в который полностью помещается фигура*



} *Примечание:*

*1.try{}…catch{}- это обработка исключения, для того, чтобы приложение не закрылось, если не будет введено значение в поле textBox1*

*В условном операторе должно быть описание области, соответствующее индивидуальному заданию (см. Практическая №3)*

Образы экрана –

|  |  |
| --- | --- |
| Сглаженные дуги и секторы | сглаженная фигура с прямоугольником и предупреждением |
|  |  |
| вычисление площади Заливка и результат  первое приближение | вычисление площади Заливка и результат  второе приближение |
|  |  |
| Скриншот папки изображение с сохраненным результатом заливки | |
|  | |

Вывод: Результаты исполнения программы и результаты контрольного теста совпали, область закрасилась, что говорит об эффективности и наглядности метода