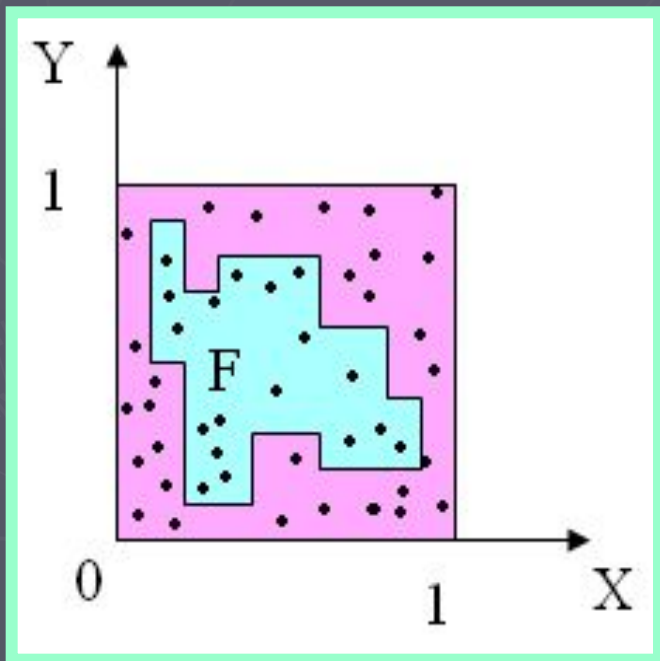


# Метод Монте-Карло

The background is a dark gray-blue gradient. In the bottom-left corner, there is a faint, light-gray illustration of a compass and a ruler. The compass needle points towards the top-left, and the ruler has markings and a dollar sign. A stylized, jagged mountain range or topographic map line pattern is visible across the bottom half of the image.

# Пример использования метода Монте-Карло при составлении информационной модели



F – плоская фигура с произвольной границей

S – площадь фигуры F (S - ?)

Будем считать, что F полностью расположена внутри единичного квадрата

Выберем в квадрате N случайных точек

N1 – количество точек, попавших в фигуру F

$$S = \lim_{N \rightarrow \infty} (N1/N)$$

# Особенности метода Монте-Карло

## 1). Простая структура вычислительного алгоритма

составляется алгоритм для выполнения одного случайного действия и повторяется  $N$  раз, причем каждый опыт не зависит от остальных

## 2). Ошибки пропорциональны $\sqrt{(D/N)}$

$D$  – некоторая постоянная

$N$  – число испытаний

# Задачи, которые решаются методом Монте-Карло

## Во-первых

метод позволяет моделировать любой процесс, на протекание которого влияют случайные факторы

## Во-вторых

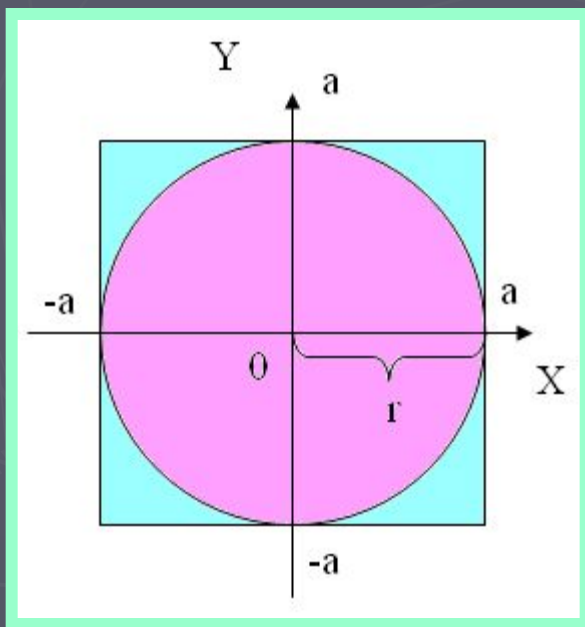
Для многих математических задач, не связанных с какими-либо случайностями можно искусственно придумать вероятностную модель, решающую эти задачи

Метод нахождения площади будет справедлив, если СЛУЧАЙНЫЕ точки будут располагаться РАВНОМЕРНО по всему квадрату

# (1 этап) Постановка задачи

Задача: Дана геометрическая фигура неправильной формы. Вычислите ее площадь. Фигура задана своей границей. Результатом решения является площадь  $S$  этой фигуры.

## (2 этап) Анализ объекта моделирования и построение информационной модели



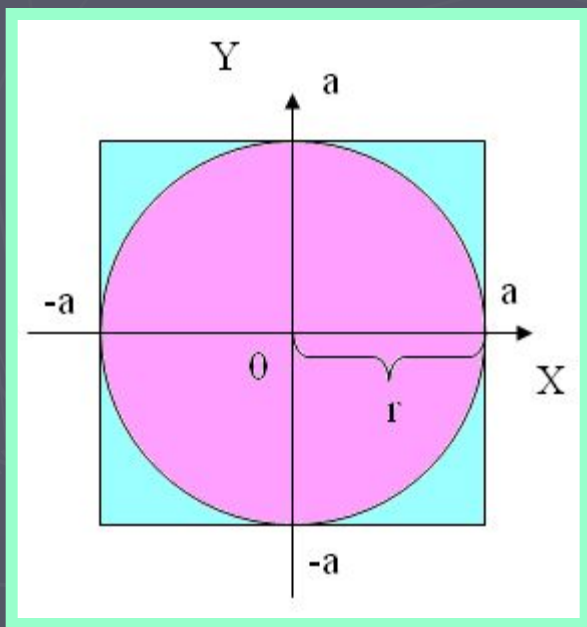
! Рассмотрим в качестве примера математическую модель нахождения площади круга радиуса  $r$ . Действовать будем по принципу первого примера. !

## (2 этап) Анализ объекта моделирования и построение информационной модели

### Шаг 1.

Примем  $a=r$ , где  $a$  – половина длины стороны квадрата, тогда  $S1$  (площадь квадрата) вычисляется по формуле

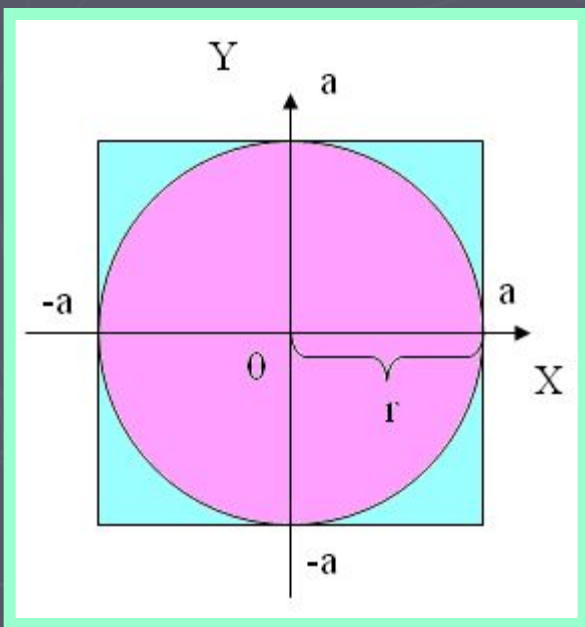
$$S1=2*a*2*a$$



## (2 этап) Анализ объекта моделирования и построение информационной модели

### Шаг 2.

Случайным образом выбираем точку, принадлежащую квадрату (задаем ее координаты, т. е  $x$  и  $y$ )



Точка принадлежит квадрату,  
если

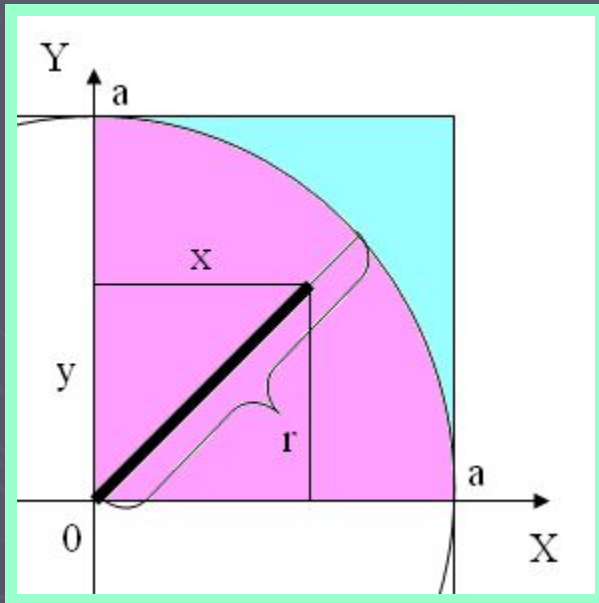
$$-a \leq x \leq a \text{ and } -a \leq y \leq a$$

Or

$$-r \leq x \leq r \text{ and } -r \leq y \leq r$$



## (2 этап) Анализ объекта моделирования и построение информационной модели



### Шаг 3.

Точка принадлежит кругу,  
если справедливо неравен-  
ство:

$$x^2 + y^2 \leq r^2$$

Or

$$x^2 + y^2 \leq a^2$$

# (3-4 этапы) Алгоритмизация решения задачи и создание компьютерной модели

## Исполнитель Basic – система

Шаг 1. Ввод данных:

Число точек  $n$

Радиус круга  $r$

.....  
PRINT "Введите число бросаемых точек"

INPUT  $n$  PRINT "Введите радиус круга"

INPUT  $r$   
.....

(3-4 этапы) Алгоритмизация решения задачи и создание компьютерной модели

## Шаг 2. Запрограммируем создание последовательности случайных чисел и выбор числа из этой последовательности.

Randomize timer `создание последовательности  
`случайных чисел

Let  $x = (a - (-a)) * \text{rnd} + (-a)$

выбор следующего случ. числа из диапазона от  $-a$  до  $a$   
и присваивание его значения переменной  $x$

Let  $y = (a - (-a)) * \text{rnd} + (-a)$

- выбор следующего случ. числа из диапазона от  $-a$  до  $a$
- и присваивание его значения переменной  $x$

Т.е. точка  $(x; y)$  принадлежит квадрату со стороной  $2a$

## (3-4 этапы) Алгоритмизация решения задачи и создание компьютерной модели

**Шаг 3.** Совокупность команд, определяющих, принадлежит ли точка  $M(x;y)$  фигуре, площадь которой уже найти, оформим в виде подпрограммы-функции **Belong%**

```
FUNCTION Belong%(x AS DOUBLE,  
                y ASDOYBLE,a AS DOUBLE)  
    LET Belong%=0  
    IF  $x*x+y*y \leq a*a$  THEN  
        Belong%=1  
    END IF  
END FUNCTION
```

## (3-4 этапы) Алгоритмизация решения задачи и создание компьютерной модели

**Шаг 4.** Формирование случайным образом координат  $x$  и  $y$ , а так же вызов функции `Belong%` происходит в основной программе в цикле по  $i$  от 1 до  $n$  (количество точек). Если функция принимает значение 1 то переменная-счетчик  $m$  (в начале программы обнулен) увеличивается на 1

```
.....  
FOR i= 1 to n  
  LET x=2*a*rnd-a  
  LET y=2*a*rnd-a  
  IF Belong%((x),(y),(a))=1 then  
    LET m=m+1  
  END IF  
NEXT i  
.....
```

## (3-4 этапы) Алгоритмизация решения задачи и создание компьютерной модели

**Шаг 5.** Подсчет результата, т.е  $S = (m/n) * S1$ . Как видно из формулы, для получения  $S$  необходимо знать чему равна  $S1$  (площадь квадрата со стороной  $a=r$ ). Как упоминалось выше,  $S1 = a^2 * a^2$ . Далее – вывод результатов.

---

```
LET S1=2*a*2*a  
LET S=(m/n)*S1  
PRINT "n=";n;"S=";S
```

---

## (3-4 этапы) Алгоритмизация решения задачи и создание компьютерной модели

### Комментарии:

- I** В данной задаче так же возможно провести подсчет времени, в течение которого выполнялись все действия программы. Это осуществляется следующим образом:
- 1) После запуска **RANDOMIZE TIMER** переменной **time1** присваивается значение **timer**
  - 2) После окончания работы цикла по **I** от 1 до **n** переменной **time2** присваивается значение **timer**
  - 3) Находится разница: Время счета  $time2 - time1$
- II** Программу можно дополнить имитацией бросания точек в квадрат

## (6-7 этапы) Вычислительный эксперимент. Анализ результатов эксперимента

Для проведения эксперимента возьмем радиус круга  $r=1$ . В этом случае площадь круга равна числу  $\pi$

Результаты эксперимента:

N	100	200	...	100000	...
П	3,16	3,152	...	3,13836	...



**Всё!**

